Autonomous Learning for Robots in the Context of Brain-Body Interactions

by

Ali Marjaninejad

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Biomedical Engineering)

August 2021

Acknowledgements

I would like to express my strongest gratitude to my mentor and advisor, Dr. Francisco J. Valero-Cuevas, who was there for me during this adventurous journey to share his insight and provide his support, whenever I needed it. He taught me, through example, how to be a good engineer, a good scientist, and most importantly, a good human being. I don't know how well I have learned the latter but I am sure I am doing better than where I was standing before joining this program and that is still progress.

I would also like to thank my committee members. Dr. James Finley, who I had the pleasure of meeting and working with even before I come to USC. His professional skills and welcoming attitude had played an important role in me deciding to join USC and work on this line of research. Dr. Stefanos Nikolaidis, who would always pump us with positive energy and motivation whenever we bumped into him on the 4^{th} floor of the Ronald Tutor Hall (RTH) building and who always finds a genius way to both point out to the strengths of your work and give you insight on how to improve on the shortcomings.

I would also like to thank my mentors and collaborators Dr. Gerald E. Loeb, Dr. Jie Tan, and Dr. Alice C. Parker, who have provided me with valuable feedback and guidance regarding my PhD work. Also, I would like to thank Dr. Hava T. Siegelmann whose vision and energy launching the L2M (Lifelong-learning Machines) program at DARPA enabled much of my work. That program introduced me to a world-wide community of elite scientists who care deeply about autonomous learning, are leading this revolution, and from whom I learned so much.

Last but not least, I would like to thank my family and friends. I would like to especially thank my mother, whose sacrifices made it possible for me to be where I am right now. I would also like to thank all Brain-Body Dynamics Laboratory (Valero lab) members, especially Dr. Brian A. Cohn, Suraj Chakravarthi Raja, Darío Urbina-Meléndez, Dr. Kian Jalaleddini, Dr. Daniel A. Hagen, Dr. Jasmin A. Berry, Dr. Christopher Laine, Romina Mir, and Pegah (Parmita) Ojaghi, from whom I have learned a lot during these years and with whom I believe I have grown to be a better person. Also, I would like to thank all my non Brain-Body Dynamics Laboratory friends as well. Especially Dr. Zumra Peksaglam, Hydari Masuma Begum, Cecilia Brown, Dr. Amirhosein Mousavi, and Dr. Alireza Delfarah, who have made me truly enjoy all my PhD years and form some of the best memories of my life. I truly appreciate all of the people in my life and feel extremely lucky to have such an amazing circle of support.

Table of Contents

List Of Tables

List Of Figures

Abstract

Robots will become ubiquitously useful only when they can learn how to perform different tasks in an autonomous, data-efficient, and generalizable (across different body structures or environments) way. Biological systems, especially vertebrates, set a great example: they learn how to perform multiple tasks after a relatively short and sparse trial-and-error process even if their bodies are particularly difficult to control.

Vertebrate bodies are hard to control (at least from the engineering perspective) because they have musculotendon-based actuation that makes them simultaneously nonlinear, under-determined and over-determined. However, this anatomy provides very important benefits such as the ability to have the center of the mass closer to the main body. Tendon-driven actuation plays an important role in the enviable functional versatility that vertebrates possess.

It is possible to improve on the current state of robotics by finding inspiration from useful mechanisms in both anatomy and controls in vertebrates. Namely, robots can and should benefit from the principles of tendon-driven structures to efficiently and autonomously learn how to control their bodies using sparse sampling, modular and hierarchical control structures and artificial neural networks that map sensory inputs to actuation signals.

In this dissertation, I have provided a new approach that enables robots to start learning without an explicit model of their body or the environment (and therefore do not need to bridge the Sim-to-Real gap), learn from limited-experience, and adapt on the fly. This approach enables model-agnostic autonomy in robots as they can learn on the spot directly from interactions with the physics of the world, while equipping them with many of the benefits that tendon-driven anatomies provide.

Chapter 1

Introduction

1.1 Current state-of-the-art in robotics and its limitations

We would like robots to participate in everyday tasks in the same environments and settings that we humans live and interact in. However most robots are currently limited to specific tasks and structured environments. Although many useful robots work in numerous instances these days, they are mostly limited to tasks with repetitive routines, structured environments, or predictable situations.

This is because most of the current robots are limited in both their learning and adaptation capabilities as well as their physical structures. The goal of robots being able to contribute to everyday tasks and work side-by-side with humans will be achievable when (i) they are able to learn and adapt quickly (i.e., in a data-efficient way) and can generalize to tasks and situations that they have not been exposed to (or have a minimal prior about) and (ii) they perform in a safe and energy efficient way.

Most of the current machine learning algorithms that deal with such autonomous learning depend on large amounts of data (e.g., supervised-learning), extensive trial-and-error processes (e.g., reinforcement learning), or a comprehensive understanding encoded in an accurate model of the system or the environment [81, 4, 65, 109, 113, 83, 93, 36, 22, 9, 67, 58].

Although these methods can yield impressive individual results in simulation or structured environments, they fail to generalize in unstructured real-world applications—where accessing large databases or performing extensive trial-and-error processes is usually not feasible in a short amount of time, and carries high risk and opportunity costs [90, 72, 42]. Autonomous learning systems also need to be robust to changes in their own structure, or errors in their modelling parameters. They must learn how to control themselves even when there is no model, or the model is not accurate or has been thoroughly changed (damage, payloads, wear and tear, etc.) [9, 72, 90, 91, 89].

Moreover, for a system to be able to contribute in everyday tasks that incorporate unmodeled situations and uncertainty, being able to generalize to new situations and adapt on the fly is a critical factor [89, 116]. Therefore, model-agnostic data-efficient learning algorithms that can adapt on the fly are of exceptional importance to make it feasible for robots to contribute to real-world challenges in unstructured environments.

Robots that use tendon-driven structures are a particular case of high interest. They will allow us to study and consider future generations of biologically inspired robots because they can provide unique kinematic capabilities, energetic efficiency, and safety benefits [148, 92]. Tendon-driven systems are flexible in their design as they can have multiple actuators and different tendon routings. Different tendon routings can produce different moment arm values across joints, which in turn can create trade-offs between the endpoint forces and velocities[148, 87, 86, 92]. Moreover, flexibility in actuator placement that acts at a distance via tendons is another benefit that tendon-driven systems provide. Thanks to this feature, actuators can be moved toward the center of mass and be placed at the locations that do not experience much movement. This increases the agility and energy efficiency of the system as it is faced with smaller inertia in its moving parts (i.e., the limbs). Moreover, thanks to the low inertia on the moving parts, tendon-driven systems would be a safer choice for robots that work next to humans. By using elastic tendons or backdrivable motors, tendon-driven systems can also be more forgiving when making or breaking contact with objects and the environment and therefore simplify control and pose less of a risk to humans near them.

Tendon-driven systems, however, are known to be particularly difficult to model and control as their control is simultaneously under- and over-determined. The production of net joint torques can be under-determined (more muscle than joints) if the torque at a *single* joint is determined by *all* the tendons crossing that joint (even when interaction torques are ignored). The control of tendon excursions can be over-determined because the angles of a *single* joint determine the excursions of *all* tendons crossing it. [148, 92].

Clearly, tendon-driven robots can–and are predominantly designed–to have only two tendons per joint. But this is simply a way to implement direct rotational actuation (the equivalent of having a dedicated rotational motor at each joint). However, we are interested in utilizing the added benefits that more elaborate tendon routing strategies can provide [148, 92]. Moreover, we are interested in *how* Nature can–and predominantly has–evolved systems with different muscle-to-joint ratios (i.e., from a few more to many more muscles than joints) and complex tendon routing strategies. We can then use the lessons learned from those biological architectures and control approaches to endow engineered robots with the enviable versatility of animals.

1.2 Contributions of the current work to the field

The limitations and needs mentioned in the previous section are the reason why this thesis is focused on an algorithm that can learn quickly (i.e., be data-efficient) on an arbitrary body (tendon-driven or not) and be able to generalize to new situations and adapt on the fly, without a need to be pre-trained in simulation or having access to a detailed kinematic model (i.e., be model-agnostic) and can work in tendon-driven structures (i.e, apply to bio-robotics).

At its core, the General-to-Particular (G2P) algorithm that I demonstrate first explores the *general* relationship between the actuators and system kinematics through sparse sampling and then creates an initial mapping to be applied to a *particular* task—similar to play(a combination of apparently random play followed by goal-driven practice) in the development and learning in animals including humans. Next, for any given task, the algorithm will continually collect new task-specific data, combine it with the data it has collected so far, and refine its mapping using the concatenated database. We have tested the G2P algorithm in a real-world physical tendon-driven system and it has shown that it can learn to perform functional tasks within minutes—compared to hours, days and months worth of data needed by other state-of-the-art algorithms [132, 4, 100, 153, 137, 31].

1.3 Consequences of this work

This work moves us one step closer to achieve the enviable learning and adaptation capabilities seen in biological systems, as well as their exceptional agility and energy efficiency [90, 89, 148, 92, 17, 94, 61]. The proposed model-agnostic approach enables learning using little or no information on the kinematic model of the system. This is particularly important for complex structures such as non-rigid robots (soft robotics, tendon-driven systems, hybrid soft-hard robots, etc.) where it is very difficult to model and simulate the system. Moreover, this model-agnostic approach can also enable autonomous design to be a part of the robotic co-evolution of controller and body, and therefore allow robots to autonomously change and optimize their 'anatomy' and control architecture simultaneously. In addition, the proposed data-efficient learning approach and on-the-fly adaptation enables learning in the 'Real-World' without extensive (and expensive) trial-and-error processes which will pave the way for robots to be used in unpredictable and unstructured environments such as search-and-rescue in unknown environments or after natural disasters, and planetary exploration. In addition, the fact that this approach can work with tendon-driven systems and elastic elements enables the emergence of energy efficient and agile biologically-inspired or totally innovative robotic anatomies.

The following five chapters forming this dissertation are organised as follows: In Chapter 2, a complete over-view of tendon-driven systems is provided along with a deeper insight into their benefits and challenges using both mathematical formulation and computational models. Chapter 3 introduces the G2P algorithm, which enables autonomous learning in tendon-driven systems using limited-experience and show cases its effectiveness in controlling a 3-tendon, 2-joint physical limb to produce functional movements. Chapter 4 evaluates the G2P algorithm in the presence of parallel elastic element and shows how it can work with, and even exploit, the passive characteristics of the system's physics to accelerate learning and enhance performance. Chapter 5 provides a closed-loop approach to utilize kinematic feedback to further improve movement accuracy even in the presence of changes in the task. Lastly, Chapter 6 expands the G2P algorithm to a complete quadruped (in simulation) and studies the forward and backward generalizability across a set of task and the effects of tactile sensory on its lifelong learning capabilities without catastrophic forgetting. Chapters 2-5 are fully or partially published in [92], [90], [91], and [89], respectively.

1.4 Limitations of this work

We have only studied robots whose limbs have 2 joints and 3 tendons (N+1 design; minimum number of muscles needed to control N joints). However, the scalability of this approach to systems with more muscles, joints, and kinematic layouts is an interesting question that should be addressed in future work.

Moreover, here we mainly focused in kinematic control and did not go deeply into higher-level learning, nor into different approaches to combine these meta- and higher-level learning approaches with the proposed G2P algorithm. We studied how our approach can be combined with reinforcement learning in a hierarchical fashion to perform the functional task of propelling a treadmill (3). Exploring other hierarchical learning architectures and approaches that can address more complicated tasks—and study performance across a wider set of different tasks—is another interesting research topic that can be addressed in future work. In addition, here we studied the adaptability of the system to different dynamical complexities but did not study how it can detect and adapt to major changes in the body (e.g., caused by damage) which can also be the aim for a potential future investigation.

Furthermore, we have mainly used a simple multi-layer perception Artificial Neural Network (ANN). Studying different ANN structures can lead to increased performance and data-efficiency. This can be especially interesting for the quadruped design when there are more options on how to develop the ANN structure (e.g., an independent ANN for each leg, a single ANN for all legs, or a hybrid approach where ANNs are mainly independent but they would be connected in some levels). Nature seems to implement various levels of connectivity among local and global networks in its hierarchical and distributed architectures [88, 41, 39, 96, 60, 147], which we should explore in the future.

Lastly, although we tested our approach in both simulation and hardware, we have limited the quadruped studied here to simulation only, for now. Having the quadruped system in hardware would be an important next step (on which we are working) to test the full potential of this approach and study its real-world performance.

Chapter 2

Should anthropomorphic systems be“redundant”?

Ali Marjaninejad[1], Francisco J. Valero-Cuevas,[1,2]

[1]Department of Biomedical Engineering, University of Southern California, Los Angeles, CA, USA

[2]Division of Biokinesiology and Physical Therapy, University of Southern California, Los Angeles, CA, USA

2.1 Chapter summary

In this chapter, we explore the conceptual design and implementation of muscle redundancy and kinematic redundancy for anthropomorphic robots from three perspectives: (i) The control of tendon-driven systems, (ii) How the number of muscles define functional capabilities, and (iii) How too few synergies can be detrimental to functional versatility.

Historically, roboticists prefer either rotational actuators located at each joint (i.e., rotational degree-of-freedom, DOF), or few linear actuators (i.e., two dedicated "muscles" per joint) for tendon-driven robots. In contrast, biological limbs have evolved to include "too many" muscles [148], which are thought to unnecessarily complicate their anatomy and control. The question, then, is why has evolution converged on these apparently under-determined (or 'redundant') solutions? If we really have extra muscles, then which muscle would you give up? By taking a formal mathematical approach to the control of tendons—which is the actual problem that confronts the nervous system—we have proposed a resolution to this apparent paradox by proposing that vertebrates may have, in fact, barely enough muscles to meet the numerous physical constraints for ecological functions (as opposed to simple laboratory tasks) [148, 78]. This approach can be called Feasibility Theory, which describes how the anatomy of the system, and the constraints defining the task define the set of feasible actions the system can produce. The role of the (neural or engineered) controller is then, to find ways to use the mechanical capabilities of the combined controller-plant system to the fullest [148].

Similarly, the effective mechanical design of a robotic limb, at a minimum, requires controllability (i.e., enough control degrees of freedom, or muscles) to produce arbitrary forces and movements (i.e., changes of state [112]). Force and movement capabilities have distinct governing equations and are, in fact, in competition with one another (e.g., a see-saw demonstrates, as per the Law of Conservation of Energy, how producing higher forces is associated with lower velocities and vice versa). Therefore, we explored the potential evolutionary pressures that may have shaped vertebrate limbs by evaluating how the number of muscles affects the competing demands to produce endpoint forces and velocities.

A related concept that cuts across biological and robotic systems is the idea that the kinematics and kinetics of a wide variety of actions exhibit a low-dimensional structure that can be approximated with a few principal components (sometimes called descriptive synergies [10, 130]). This has been taken to mean that a few degrees of freedom suffice to produce versatile behavior in the real world. However, the fine behavioral details that distinguish different actions are, by definition, not captured by the commonalities among them. Thus, versatility in the real world likely depends on recognizing and executing fine distinctions among actions; which implies that more degrees of freedom of control are critical for true functional versatility.

These three independent arguments support the perspective that creating anthropomorphic systems requires apparently redundant structures, because only then can they truly execute a wide variety of real-world tasks. In addition, we also present an open-access MATLAB toolbox that allows users from different backgrounds to explore these concepts in detail. We believe this new perspective will improve the conceptualization, understanding, and design of anthropomorphic systems.

2.1.0.1 Author Contribution

A. M. has designed experiments, wrote the code, run simulations and written the first draft of the manuscript. F.V-C. has provided guidance and feedback to the manuscript and during each step of the experiment design and interpreting the results.

2.2 Mechanical and neural foundations of Feasibility Theory

In this section, we provide fundamental concepts required to study neuromechanical systems, which will set fundamentals for the following sections of this chapter. We begin with limb kinematics, providing a common conceptual language to muscle mechanics. Next, we introduce motor control and feasible movements of tendon driven limbs to show how tendon-driven systems are in fact over-determined. Due to the inherent properties of biological muscles, the nervous system is likely not as redundant as when considering the force control problem in isolation. Moreover, we have described how different task constraints (i.e., the mechanical definition of the task) naturally limit feasible actions. Thus, each additional muscle adds an additional control DOF—and therefore the ability to meet more simultaneous functional constraints and produce a wider variety of tasks—which is the origin of versatility. Many of the materials and concepts of this section are summarized or first introduced in [148].

2.2.1 Limb kinematics and limb mechanics

We first start with limb kinematics, which characterize the motions and positions of rigid bodies, regardless of the forces which produce them. In this chapter, in order to simplify the governing equations, we consider limbs as rigid bodies. Subsequently, we introduce equations for limb mechanics which involve limb kinematics as well as the forces and torques that interact with the limb.

2.2.1.1 Limb kinematics

We first define a limb as a set of connected links and hinges. The endpoint of a multi-joint limb is defined by the homogeneous transformation matrix $\mathrm{T}_{base}^{endpoint}$. $\mathrm{T}_{base}^{endpoint}$ can also be written as the multiplication of the DOF transformation matrices:

$$T_{base}^{endpoint} = T_0^N = T_0^1 T_1^2 \cdots T_{N-2}^{N-1} T_{N-1}^N \tag{2.1}$$

where each transformation matrix is defined as:

$$T_i^j = \left\{ \begin{array}{cc} R_i^j & \mathbf{p}_{i,j} \\ 0\,0\,0 & 1 \end{array} \right\} \tag{2.2}$$

In the above equation, R_i^j represents the rotation matrix and $\boldsymbol{p}_{i,j}$ represents the displacement for each DOF [148]. A schematic representation of the system for Eq. 2.1 is plotted in Fig. 2.1. Furthermore, the forward kinematic model (also known as the geometric model), G(q), for a planar system (in two-dimensional space) is defined as follows (given that a rigid body on the plane has three degrees of freedom defining its location and orientation):

$$G(q) = \begin{pmatrix} displacementinthedirectionofi_0 \\ isplacementinthedirectionofj_0 \\ rotationabouttheк_0axis \end{pmatrix} = \begin{pmatrix} x \\ y \\ \alpha \end{pmatrix} \tag{2.3}$$

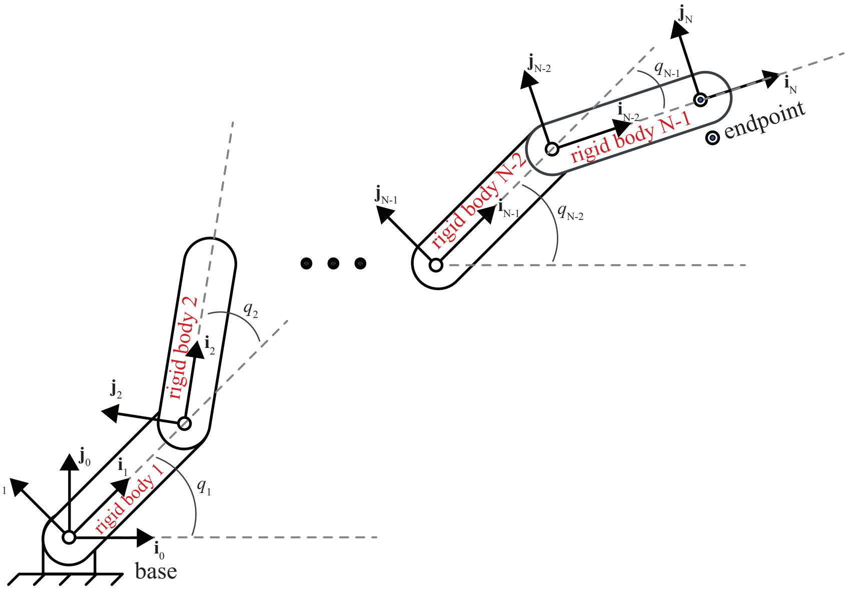


Figure 2.1: A schematic representation of a simple multi-link system. (Reproduced, with permission, from [148])

For non-planar limbs, displacement in the direction of k_0, rotation about the i_0, and rotation about the j_0, will also need to be included in G(q) [148]. The endpoint velocities are obtained by differentiating G(q) with respect to time:

$$\dot{G}(\mathbf{q}) = \frac{dG(\mathbf{q})}{dt} = \frac{\partial G(\mathbf{q})}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} = \frac{\partial G(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{pmatrix} \tag{2.4}$$

We call $\frac{\partial G(\mathbf{q})}{\partial \mathbf{q}}$ as the Jacobian matrix. For a limb with N degrees of freedom, it is defined as:

$$\frac{\partial G(\mathbf{q})}{\partial \mathbf{q}} = J(\mathbf{q}) = \begin{bmatrix} \frac{\partial G_x(\mathbf{q})}{\partial q_1} & \frac{\partial G_x(\mathbf{q})}{\partial q_2} & \dots & \frac{\partial G_x(\mathbf{q})}{\partial q_N} \\ \frac{\partial G_y(\mathbf{q})}{\partial q_1} & \frac{\partial G_y(\mathbf{q})}{\partial q_2} & \dots & \frac{\partial G_y(\mathbf{q})}{\partial q_N} \\ \frac{\partial G_\alpha(\mathbf{q})}{\partial q_1} & \frac{\partial G_\alpha(\mathbf{q})}{\partial q_2} & \dots & \frac{\partial G_\alpha(\mathbf{q})}{\partial q_N} \end{bmatrix} \tag{2.5}$$

2.2.1.2 Limb mechanics

As mentioned earlier, limb mechanics involve limb kinematics as well as the forces and torques the limb can produce. In this section, we will explain how to relate joint torques to endpoint forces using limb kinematics. Let's begin by defining the internal and external work for the two DOF systems shown on Fig. 2.2 as:

$$External work = \mathbf{f} \cdot \Delta \mathbf{x} \tag{2.6}$$

$$Internal work = \tau \cdot \Delta \mathbf{q} \tag{2.7}$$

where **f** and $\Delta\mathbf{x}$ are scalar values representing endpoint force and endpoint velocity, while τ and $\Delta\mathbf{q}$ are torque and joint angle rotation vectors. Please note that, depending on the kinematic DOFs of the system, the endpoint can also produce a torque ($\tau_{endpoint}$), whose external work would be it times the rotation of the endpoint. For the sake of simplicity, here we present the derivation only considering endpoint forces. Following with Eq. 2.7 and Eq. 2.8, from the conservation of energy law, we have:

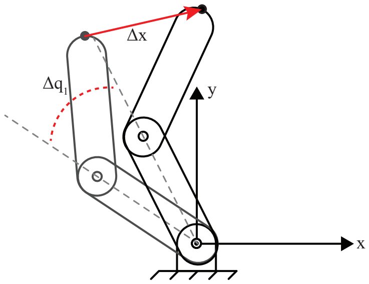


Figure 2.2: An illustration of the geometric relationship between the endpoint displacement ($\Delta\mathbf{x}$) and the rotation of the first joint (Δq_1 in a 2-DOF limb). (Reproduced, with permission, from [148])

$$\mathbf{f}.\Delta\mathbf{x} = \tau.\Delta\mathbf{q} \tag{2.8}$$

Changing the dot product in Eq. 2.8 to its equivalent inner product while substituting Δ for both **x** and **q** with derivatives, we have:

$$\mathbf{f}^T\dot{\mathbf{x}} = \tau^T\dot{\mathbf{q}} \tag{2.9}$$

Using Eqs. 2.4 and 2.5, we rewrite Eq. 2.8 as:

$$\mathbf{f}^T J(\mathbf{q})\dot{\mathbf{q}} = \tau^T \dot{\mathbf{q}} \tag{2.10}$$

Eliminating $\dot{\mathbf{q}}$ from both sides of the equation leads to Eqs. 2.11 and 2.12, which define the relationship between the joint torques and endpoint forces.

$$\mathbf{f}^T J(\mathbf{q}) = \tau^T \tag{2.11}$$

$$\mathbf{f}^T = J(\mathbf{q})^{-1} \tau^T \tag{2.12}$$

Extensions of this concept for 3-dimensional space with detailed examples are provided in [148].

2.2.1.3 Tendon-driven limb mechanics

Most robotic limbs are driven by either rotational or linear actuators that drive each kinematic DOF [148]. In the robotics literature, the so-called torque-driven formulation assumes symmetric actuators. That is, equal torque capabilities in both clockwise and counterclockwise directions.

In the so-called tendon-driven systems, actuators are connected to the limbs using strings, cables, or tendons. It is clear that these actuators can only pull (and not push) on the tendons, thus the can only drive the DOF in one direction. Therefore, each DOF requires, on average, more than one actuator and symmetry of actuation is not guaranteed. In addition, tendon-driven systems are flexible because the routing of their tendon can allow one actuator to drive more than one DOF—and therefore impose correlations in actuation across DOFs [43, 111]. Moreover, the moment arms (i.e., minimal perpendicular distance between the tendon path and the center of rotation of the DOF) can be arbitrarily set within and across tendons and DOFs.

This flexibility of actuation that can be "built into" the morphology of the design, compared to torque-driven systems, introduces unique flexibility and challenges to their construction and control. To some, this means that tendon-driven systems are unnecessarily difficult to build and control. However, we and others also argue that they have much to offer [111, 75, 64, 32, 51, 53, 85]. Using tendons to apply torque to the DOFs, as opposed to having actuators directly apply torque, makes tendon-driven systems capable of remote actuation. This means that the designer can place the actuator far from the joint itself. Although making the system harder to control, flexible tendon routing can provide much more versatility and preferentially larger feasible end-point forces and velocities in directions of interest.

2.2.1.4 Tendon actuation

To explore how tendons create torque in tendon driven systems, see Fig 2.3. This simplified model illustrates a planar, one-joint limb using one muscle. The torque at the joint of this model is equivalent to the cross-product of the force and moment arm *r*:

$$\tau = r \times f_m = \|r\| \, \|f_m\| \, sin(\alpha) \tag{2.13}$$

where × represents the cross-product, f_m represents the muscle force, and α represents the angle between the force and the moment arm.

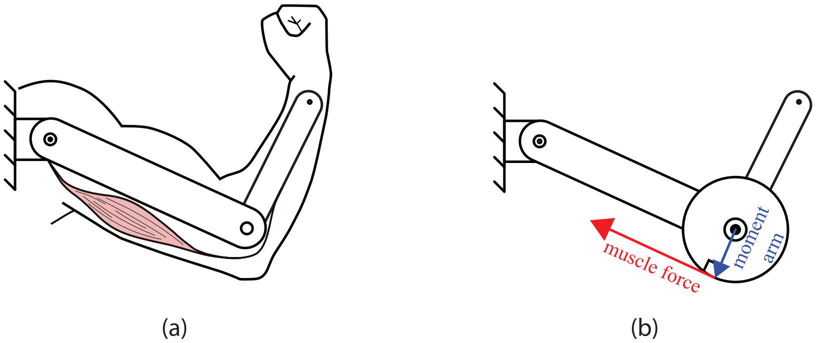


Figure 2.3: Schematic representation of a planar, two joint limb with only one muscle (a) and its equivalent simplified model (b).

As mentioned earlier, in tendon-driven systems, one actuator can exert torque in multiple DOFs. Here, we are going to study an example of such while introducing the moment arm matrix. We begin by illustrating the relationship between the torque vector (here, a vector of length two, representing the two DOFs) and muscle force vector for the two-joint planar limb shown in Fig. 2.4:

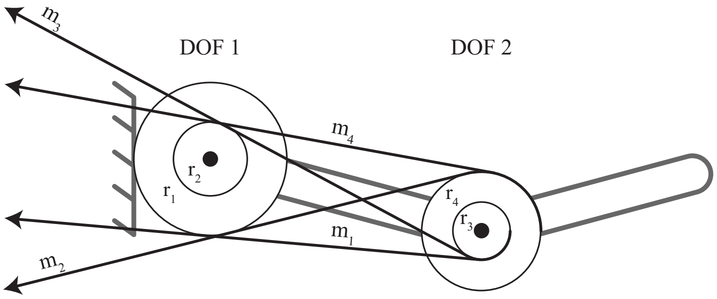


Figure 2.4: A sample two-joint limb with four muscles. (Reproduced, with permission, from [148])

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_M \end{pmatrix}_{M\times 1} = R(\mathbf{q})_{M\times N} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}_{N\times 1} \tag{2.14}$$

where R(**q**) represents the moment arm matrix, which maps the muscle forces to the joint torques. In example illustrated in Fig. 2.4, the moment arm matrix R(**q**) is defined as:

$$R(\mathbf{q}) = \begin{bmatrix} -r_1 & -r_1 & r_2 & r_2 \\ -r_3 & r_4 & -r_3 & r_4 \end{bmatrix} \tag{2.15}$$

By convention, each entry in the moment arm matrix on the i^{th} row and j^{th} column will be the coefficient which transforms the force induced by the i^{th} muscle to the torque exerted at the j^{th} joint (as seen in Eq. 2.9). The positive value of an element in this matrix means that counterclockwise (positive) torque will be applied when tension is applied to the tendon through muscle contraction (applying concentric force).

In order to relate muscle excursions to joint movements in the model shown above, we produce a set of equations. Again, by convention, we consider counterclockwise rotations as positive rotations. Following the conventions we have mentioned so far, a positive joint rotation with a positive moment arm induces a shortening in the length of its muscle and tendons and vice versa. Therefore, the set of equations relating joint angles and muscle excursions for the example provided in Fig. 2.4 will be as follows:

$$\begin{pmatrix} \partial S_1 \\ \partial S_2 \\ \partial S_3 \\ \partial S_4 \end{pmatrix}_{4\times1} = \begin{bmatrix} r_1 & r_3 \\ r_1 & r_4 \\ -r_2 & r_3 \\ -r_2 & r_4 \end{bmatrix}_{4\times2} \begin{pmatrix} \partial q_1 \\ \partial q_2 \end{pmatrix}_{2\times1} \tag{2.16}$$

Using Eq. 2.9, we can rewrite Eq. 2.10 in the general case with M joints and N muscles as:

$$\begin{pmatrix} \partial S_1 \\ \partial S_2 \\ \vdots \\ \partial S_N \end{pmatrix}_{N\times 1} = (-R_{M\times N})^T \begin{pmatrix} \partial q_1 \\ \partial q_2 \\ \vdots \\ \partial q_N \end{pmatrix}_{M\times 1} \tag{2.17}$$

Taking a closer look at Eq. 2.14 and Eq. 2.17, we see a very important distinction. Equation 2.14 is under-determined, meaning that there is more than one solution in the force space to achieve the desired set of torques. This is one of the main reasons that neuromuscular systems are thought to be redundant. However, looking at Eq. 2.17, we notice that this equation is over-determined, meaning there is, at maximum, only one set of values for the joint angles fulfilling this set of equations [148, 43, 150]. In other words, you cannot contract and shorten a specific muscle without having changes in the lengths of other muscles connecting to or passing through the same joint. It illustrates that our nervous system must consider a complex variety of constraints when pulling a tendon. Failure in fulfilling the requirements of this complex control problem, especially failure in relaxing muscles that are being lengthened as a result of joint rotations, might disrupt movement or injure muscles or tendons [43, 126]. Note that this over-determined case only arises when treating limbs as tendon-driven systems. This is one of the important, yet mostly overlooked aspects of robotics today.

2.2.2 Motor control of tendon driven limbs and Feasibility Theory

In this section, we describe the tendon-driven system that the nervous system faces. First, we present a conceptual framework where one can think of neural commands as being a high-dimensional activation vector that is mapped into lower-dimensional "spaces", that capture its transformation into endpoint forces. Next, we describe Feasibility Theory, which defines how different constraints can limit the feasible activations in different spaces (activation, muscle force, torque, and end-point force spaces). If we are looking for a versatile system to deal with day to day activities, then a larger number of DOFs are required as constraints are added.

2.2.2.1 Motor control of tendon driven limbs

Although the nervous system activates muscles through the recruitment of motor neurons and modulation of their firing rates, we can, without loss of generality, simplify the problem by assigning a value between 0 and 1 to the activation level of each muscle, where 0 represents complete inactivation and 1 represents maximal activation. The activation vector α is described as Eq. 2.18 for an N muscle system:

$$\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix}, \ 0 \leq \alpha_i \leq 1 \ for \ i = 1, \ldots, N \tag{2.18}$$

where α_i is the activation value for the i^{th} muscle. Now the set of the generated muscle forces at a particular moment and at a particular activation level can be defined as:

$$\mathbf{f_m} = F_0(\mathbf{l_m}, \mathbf{v_m})\alpha \tag{2.19}$$

where F_0 is the diagonal matrix. Each element on the main diagonal of this matrix will represent the maximum force that the corresponding muscle can exert. These diagonal values depend on many factors such as muscle architecture, pennation angle, physiological cross-sectional area, as well as the fiber length (l_m) and velocity (v_m) of the muscle at every time point. Now, we rewrite Eq. 2.14 by substituting the force vector from Eq. 2.19, which leads to the following equation:

$$\tau_{\mathbf{M}\times\mathbf{1}} = R(\mathbf{q}_{M\times N})\, F_0(\mathbf{l_m}, \mathbf{v_m})_{N\times N}\, \alpha_{N\times 1} \tag{2.20}$$

Eq. 2.20 shows that the control of joint torques in tendon-driven limbs is an underdetermined set of equations. However, adding a cost function which needs to be minimized (e.g. the total sum of the activation values), will force this set of equations to have fewer possible solutions (or even just one).

Considering the fact that maximum muscle force values (the diagonal values on the F_0 matrix) are also functions of muscle length and velocity, we see that discovering the activation values which result in a desired set of joint torques (Eq. 2.20) is difficult. In fact, it will apply constraints to the solutions that the nervous system can produce. We discuss the effects of these constraints in greater detail in the following sub-section, which explains how our nervous system faces a much more complex control problem than initially hypothesized in the literature [54].

Reconsidering the set of tendon excursions, we can rewrite Eq. 2.17 in terms of the muscle length vector, $\mathbf{l_m}$, and muscle velocity vector, $\mathbf{v_m}$ (Eq. 2.21 and Eq. 2.22).

$$(\delta \mathbf{l}_m)_{N\times 1} = (-R_{M\times N})^T \delta \mathbf{q}_{M\times 1} \tag{2.21}$$

$$(\mathbf{v}_m)_{N\times 1} = (-R_{M\times N})^T \dot{\mathbf{q}}_{M\times 1} \tag{2.22}$$

These last two sets of equations illustrate how control of tendon excursions is an over-determined problem (i.e. there is, at most, only one set of solutions for it). Therefore, our nervous system faces a biomechanical limitation [148].

2.2.2.2 Feasibility Theory: Defining feasible sets of actions in tendon-driven limbs

In this section, we first explore how a neural activation vector is mapped into the torque space and the effect of different biomechanical parameters on this mapping. Next, we demonstrate how functional constraints can limit the feasible action spaces, therefore making it harder for the controller to find a solution within these spaces [54].

Let's assume we have three muscles, representing their activation values as a_1 to a_3. A rectangular cuboid can be used as a visual representation of these activations. This cuboid is shown in Fig. 2.5a. The corresponding muscle force vector is determined by Eq. 2.19 and represented in Fig. 5b. Now, let's say we have a two DOF joint. This assumption means that our feasible muscle force set (Fig. 2.5b), which is a three-dimensional cuboid, will be mapped into the two-dimensional feasible torque space. The amount of torque generated at each joint is determined by Eq. 2.20. The feasible torque set for this example is shown in Fig. 2.5c. We utilize the appropriate Jacobean matrix, mapping this feasible torque set to the feasible end-point force set using Eq. 2.12. The feasible force set for this example is shown in Fig. 2.5d. Looking at the different parts of Fig. 2.5 we can see the contributions of having extra muscles in each of these feasible action spaces.

For the example in Fig. 2.5, let's now assume that there is a functional constraint. This may include a certain constraint on the magnitude of the force in one axis in the end-point force space, or a certain torque in the torque space, etc. Since the torque or the end-point force spaces are two-dimensional in this example, these constraints can only be points or lines in these spaces. However, when we track them back into the muscle activation space (or muscle force space), they can be points, lines, or planes (or hyper-planes in more than three dimensional spaces) since all the transformations are linear. Say we have defined a constraint whose representation in the muscle activation space is a plane (i.e. the constraint plane). The new feasible activation set now is the intersection of the feasible activation set, without any constraints, with this constraint plane. This new feasible activation set is shown in Fig. 2.6a. If more constraints are added, the feasible muscle activation set will lose even more dimensions and might become a line, a point, or an empty set (Fig. 2.6b-d). This shows that while creating a specific amount of end-point force or torque in a joint can be an under-determined problem, functional constraints and feasible action spaces (as well as the mapping functions between these spaces) can limit the abilities of our neuromechanical system to a great extent [148, 54]. Therefore, the control problem which the nervous system faces is a very complicated one.

This suggests that having extra muscles is not redundant, but is a necessary requirement for versatility. Extra sets of muscles will increase the DOF of the system while enabling the nervous system to find solutions for different sets of problems we face daily.

2.3 An evolutionary fitness approach to the relationship between the number of muscles and versatility

Throughout the years, many have wondered why the anatomy of vertebrates has evolved to include a seemingly redundant number of muscles. Here we show how extra muscles can enhance mechanical versatility using an evolutionary approach while clarifying why muscle redundancy is not a comprehensive belief. We begin our study by observing how our biomechanical tendon-driven model performs a set of specified kinetic tasks while changing the number of muscles. We study the optimal number of muscles with

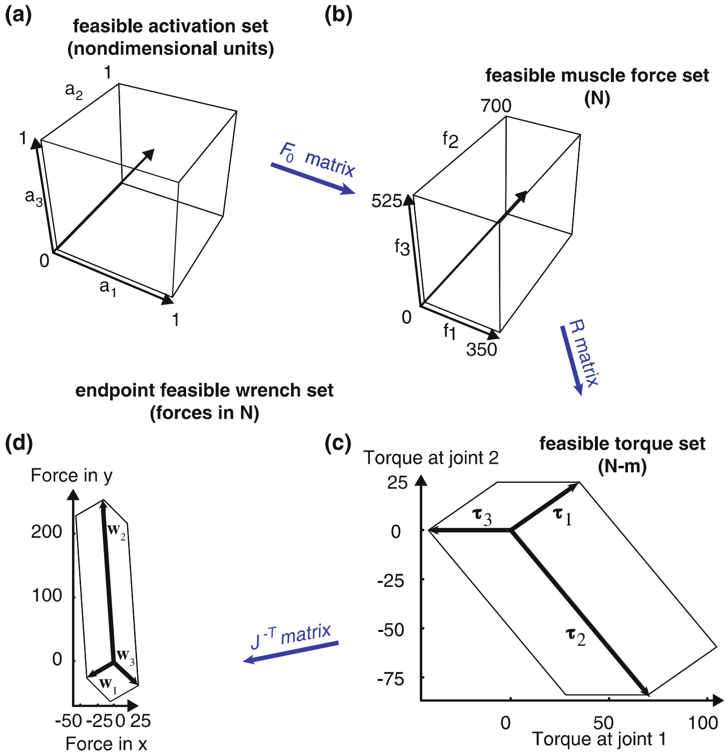


Figure 2.5: The representation of the feasible actions in different spaces for the toy example discussed in the text. (a) the feasible set of activations. (b) the feasible set of muscle forces. (c) the feasible set of joint torques. (d) the feasible set of end-point forces. (Reprinted, with permission, from [148])

three main fitness functions. Namely, the Effectiveness, Agility, and Phenotypical cost. During all tasks, the goal was to apply maximum force in a specific direction. We found the optimal activation values as a function of task restraints using a linear optimization algorithm.

2.3.1 Background

Muscle redundancy has been discussed extensively ever since the earliest neuromuscular studies. Evolution from the earliest ape species, *Nakalipithecus nakayamai*, to modern

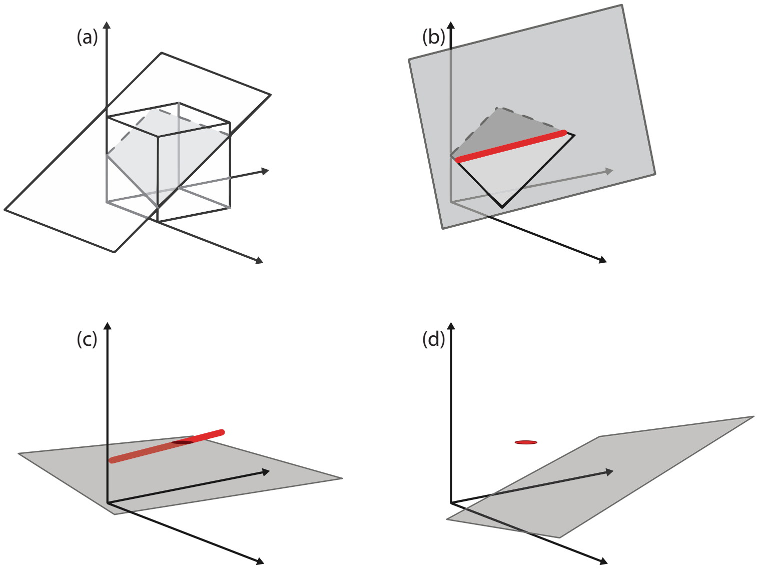


Figure 2.6: The effects of adding functional constraints on the feasible actions' spaces. More and more constraints are applied as we move from (a) to (d). (Reproduced, with permission, from [148])

Homo Sapiens, exemplifies the growth and development of muscles, enabling our species to perform multiple tasks. There are certain questions that resurface each time someone tries to explore this field. Why do we have so many muscles despite the limited degrees of freedom in our limbs or fingers? Why do we have that specific number of muscles? What are the costs and benefits of having this set of muscles?

In this section, we explore answers to these questions with an evolutionary fitness approach. We study how extra sets of muscles affect Effectiveness, Agility, and Phenotypical cost (we will describe each later in this section). In addition to this, we explore how decreasing the number of muscles affects performance in different tasks. In this study, we have used a 2-DOF arm model with three different muscle sets.

2.3.2 Method

We first explain our model and assumptions. Next, we describe the simulated tasks used in this study. Finally, we introduce our fitness functions for each of the elements mentioned earlier as well as the Overall Fitness, which is the weighted linear combination of all the individual fitness functions. Note that we use the term "fitness" in the general sense where it is not necessarily tied to a single specific cost function. Rather, fitness in the biological sense indicates the ability to meet current multi-dimensional requirements and perform well in a given environment. For the sake of simplicity, we define fitness as the ability to meet a compound cost function—but other cost functions may also be suitable depending on the functional goals at hand.

2.3.2.1 Model

We begin with a simplified model of the human arm with a two DOF planar structure. One can also generalize this model to other body parts with similar structure e.g. fingers. We select four postures based on common tasks performed by the arm, as shown in Fig. 2.7. Each posture was held static while force was maximized in four directions: upward, downward, frontward, and backward.

Three muscle sets were designed to compare the effects of varying the number of muscles. These muscle sets, shown in Fig. 2.8, were designed with the intent of recreating a model with a realistic set of arm muscles (although this model is constrained to two-dimensional space) as well as models with fewer or more muscles than a real arm. Across muscle sets, the number of muscles was decreased while keeping the same original routing configuration of the previous muscle set. This was done to compare the effects of decreasing or increasing the number of muscles only. The moment arms were estimated with reference to [148].

We selected 3, 7, and 14 as the number of muscles which are the minimum number of muscles for a 2-DOF system, the real number of muscles in a human arm and two times of the number of muscles in a human arm, respectively. In the Monte Carlo analysis, the moment arm values were varied by $\pm 20\%$ over 100 simulations to test the sensitivity of the results to these values.

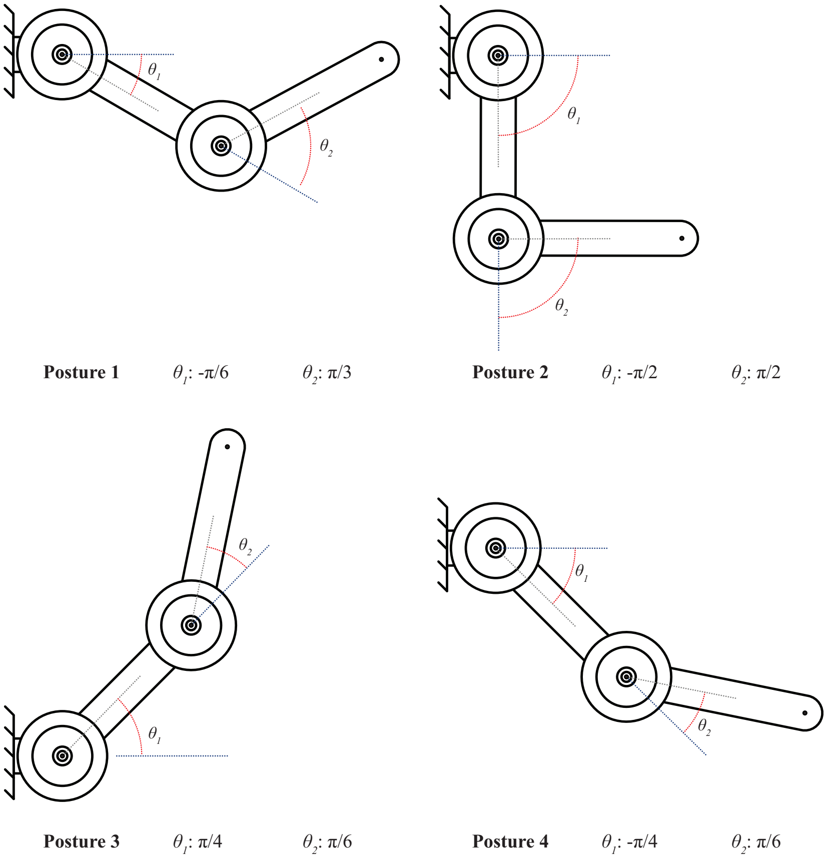
Model assumptions In this study, all muscle lengths and maximum muscle forces were assumed to be equal enabling us to address only a specific set of questions. We study how the number of muscles affects a specific set of outputs in the absence of other variables. Also, muscle forces were assumed to be independent from muscle lengths or muscle velocities.

2.3.2.2 Cost formulation

Here we describe the performance metrics studied. First, we define the Average Energy as the summed square of muscle activation values, divided by the number of muscles. Although muscle activation is a neural action, it will be proportional to physical muscle activity (and therefore requires energy) since we assumed that all muscles have equal maximum force values. Average Energy shows how hard muscles pull on average to perform the task. The Average Energy will be as defined in the quadratic Eq. 2.23.

$$Average\ Energy = \frac{1}{N}\sum_{i=1}^{N}\alpha_i^2 \tag{2.23}$$

where α_i is the activation of the i^{th} muscle (between 0 and 1) and N is the number of muscles [19, 15].



Posture 1 θ_1: -π/6 θ_2: π/3

Posture 2 θ_1: -π/2 θ_2: π/2

Posture 3 θ_1: π/4 θ_2: π/6

Posture 4 θ_1: -π/4 θ_2: π/6

Figure 2.7: Four different postures used for simulations. These postures were inspired by day to day activities.

Although Average Energy shows the total amount of energy used to perform a task, it is not the best way to calculate Effectiveness since the maximum force output of a task is also very important. Therefore, we define Effectiveness as the maximum output force divided by the Average Energy. Note that Effectiveness is distinct from efficiency as we are using it to reflect overall ability after normalizing for energy consumption. Phenotypical cost (see below) already considers metabolism.

$$Effectiveness = \frac{maximal\ force}{Average\ Energy} \tag{2.24}$$

We know that there is a limit on how fast a muscle can contract (the maximum speed for muscle excursion). We define "Agility" for each joint as the maximum rate of change in the joint (regardless of the direction), assuming that the muscle excursions for all muscles have the same upper limit. This is similar to the concept of manipulability, which considers the transformation of joint angular velocities into endpoint velocities as per the Jacobian of the limb [156]. Therefore, Agility for each joint is defined as:

$$Agility = \frac{dq}{ds} = max_{ij}(abs(\frac{1}{r_{ij}^T})) \tag{2.25}$$

where *i* is the muscle index, r_i is the corresponding moment arm value, *max* stands for maximum, and *abs* stands for absolute value. Please note that to maximize the $\frac{dq}{ds}$, r_{ij}^T

cannot be equal to zero, since this would mean that the muscle is not connected to the intended joint.

Lastly, the “Phenotypical” cost is related to the number of muscles due to the nature of muscle packaging [76]. In particular, the Phenotypical cost can consider both the cross-sectional area and volume of muscles and muscle groups. Therefore, we explored this value using the square and the cube of the number of muscles (Overall Fitness A and Overall Fitness B, respectively). The quadratic version preferentially considers metabolic and phenotypical costs associated with muscle stress and physiological cross-sectional area [157]; whereas the cubic version attempts to further penalize the complexity of vascularizing, repairing, maintaining, packaging and controlling more muscles [76].

To compute the fitness of alternative embodiments for a multi-muscle limb, we compute Overall Fitness as shown below. It is the weighted sum of the above-mentioned elements:

$$\begin{aligned} Overall\ Fitness = w_1 \times Effectiveness_N + w_2(\frac{Agility_{N,1} + Agility_{N,2}}{2}) \\ -w_3 \times Phenotypical\ cost \end{aligned} \tag{2.26}$$

where $Effectiveness_N$ and $Phenotypical\ cost_N$, are the normalized (between 0 to 1) Effectiveness and Phenotypical Cost, respectively. $Agility_{N,\ k}$ is the normalized Agility in the k^{th} joint while w_1, w_2, and w_3 are weights for $Effectiveness_N$, average $Agility_{N,\,k}$ (averaged over k) and the $Phenotypical\ Cost_N$ respectively.

We used linear programming [16] to find the solutions in the activation space for the constrained problem of maximizing the force in only the specified direction in each task.

2.3.3 Results

Figures 2.9 and 2.10 respectively demonstrate Average Energy and Effectiveness as a function of number of muscles for each posture. In addition, Agility was plotted as a function of number of muscles for each joint (Fig. 2.11). We then performed Monte Carlo analysis to determine the sensitivity of the results to the assumed values for the moment arm matrix. The results follow the same pattern regardless of the variation in values for the moment arm matrix, demonstrating that this analysis is generalizable to a large variation in moment arm values.

Our results show that increasing the number of muscles increases Effectiveness and Agility. However, it is clear that more muscles also have more Phenotypical cost. This is why we believe that there is a “sweet spot” for an optimal number of muscles, based on how much weight each of these goals have in an anthropomorphic system. These weights are set during evolution to find the optimal number of muscles to be as versatile as possible, while maintaining a reasonable Phenotypical cost. That is where the Overall Fitness, introduced in the methods section (Eq. 2.26), will be useful. The Overall Fitness plays a pivotal role by combining the weighted effect of each element and providing a single measure that needs to be optimized. Again, while adding extra muscles will increase Effectiveness (Fig. 2.10) and Agility (Fig. 11), it will also increase the Phenotypical cost. Therefore, for each system, the number of muscles for which this Overall Fitness is minimized is the optimal muscle number for that system.

By changing the weights in the Overall Fitness function, we can easily find the set of weights where 7 muscles (the real number of muscles in a human arm) are the optimal choice. As described before, these weights can be interpreted as the relative importance of each goal (Effectiveness, Agility, and low Phenotypical cost) to vertebrates, from an evolutionary point of view. By setting w_1, w_2, and w_3 to 4, 1, and 4 respectively, we have Fig. 2.12, which shows that the optimal number of muscles is 7.

Please note that there are many combinations of weights that can lead to a specific number of muscles. Similarly, any choice of cost function in the literature can be a matter of choice and preference [125]. We chose three fitness functions (Effectiveness, Agility and Phenotypical cost) to reflect the multi-dimensional nature of functional fitness. This, in fact, is best addressed as multi-objective optimization that allows espousing any one to the exclusion of others. This confronts us with the fact that any cost function is, in essence, a reflection of the multiple fitness criteria that may have been achieved, are being pursued, or are even changing in the environment. Thus, a change in environment, goals and life habits, over time, may naturally change the number and/or routing of muscles in a given anthropomorphic system. In addition, although the general patterns between simulations and real systems match, it is important to keep in mind the simplifications that were made, when comparing results from simulations to physiological recordings.

2.4 Does a minority of the variance contain a majority of information?

In this section, we explore the risks of assuming that a low-dimensional approximation suffices to capture the versatility of anthropomorphic systems. We have shown that, although a few Principal Components (PCs) can explain most of the variance in a specific movement or a set of gestures, the remaining variance can in fact contain critical details. This highlights that the reduction of DOF will come at the cost of versatility. Therefore, the fact that a few PCs capture a most of variance does not mean that anthropomorphic systems should be low-dimensional.

The problem of face recognition serves as a useful analogy. Human faces all share common features, e.g. we generally have two eyes, two ears; and the general placement of the mouth, nose, eyes and ears follows a specific pattern. However, we can recognize a particular face from among many only due to its small differences compared to the others. Similarly, when talking about hand gestures, postures and functions, the details can become very important to a specific task.

2.4.1 Background

It is known that different sets of motor actions share many commonalities. For example, a linear combination of a small set of basis vectors in a set of movements can explain large amounts of variance for each movement pattern in the set [71]. It is also true for static postures, which means a linear combination of a few basis vectors can explain large amounts of variance in a multi-DOF system; like a human hand [134]. Unfortunately, this is often over-interpreted as a sign that we have more than enough DOF, or as a sign of redundancy in anthropomorphic systems.

We have used principal component analysis (PCA) to extract the principal components (PCs) in a set of hand gestures and shown that although the first few PCs will explain a large amount of variance, all the details that make differentiating between these different movements or postures possible are present in the higher PCs. That is, although higher PCs explain less variance and are generally smaller in amplitude, they are the most important in making postures different from one another. Therefore, these extra DOF are the main contributors to versatility in anthropomorphic systems.

In this section, we further demonstrate this concept with a special focus on hand gestures. We simulate five different hand gestures, comparing and contrasting their representations in the joint angle space (19 joints). We also apply PCA to the joint angle data of these three gestures and demonstrate the effects of utilizing only the first two PCs as compared to all PCs involved.

2.4.2 Method

Grasping gestures of the hand are historically categorized into two main sets: “precision grasp” and “power grasp” [106]. In the former, the thumb and one or more of the remaining fingers will contact the object or apply force in opposition to each other. In the latter, the object will be grasped such that the palm of the hand comes into contact with it [59].

To model our distinct hand gestures, we used MuJoCo, a physics engine which provides accurate simulations for applications including robotics and biomechanics [146]. Five different hand gestures were modeled; two power grasps, two precision grasps, and a non-practical posture which we refer to as the Claw gesture. These five gestures are represented in Fig 13.

In both power grasps, the fingers opposing the thumb follow similar flexion/extension patterns in their joints. The main difference between power grasps 1 and 2 are the finger abduction values. Index, middle, ring, and pinky fingers have more space between them in power grasp 2 as compared to the power grasp 1. In precision grasp 1, only the index finger opposes the thumb while the other fingers are flexed. In precision grasp 2, the middle finger also opposes the thumb while the other fingers are less flexed. In the Claw gesture, index and ring group together with middle and pinky fingers respectively, and are opposing the thumb.

We extract 19 different joint angles for these three gestures from MuJoCo. These angles are namely Wrist PRO, Wrist UDEV, Wrist FLEX, Thumb ABD, Thumb MCP, Thumb PIP, Thumb DIP, Index ABD, Index MCP, Index PIP, Index DIP, Middle MCP, Middle PIP, Middle DIP, Ring ABD, Ring MCP, Ring PIP, Ring DIP, Pinky ABD, Pinky MCP, Pinky PIP, and Pinky DIP. We then apply PCA (similar to [134]) and compare joint angles of all five gestures before and after applying dimensionality reduction. In the reduced dimension case, we map only the first two PCs back to the joint angle space. We also calculate the Pearson's correlation coefficient for each pair of gestures in the joint angle space for before and after dimensionality reduction. The Pearson's correlation coefficient of two different gestures in the joint angle space is defined as follows:

$$Corr_{i,j} = \frac{\sum_{k=1}^{19}(angle_{i,k} - \overline{angle}_i)(angle_{j,k} - \overline{angle}_j)}{\sqrt{\sum_{k=1}^{19}(angle_{i,k} - \overline{angle}_i)^2}\sqrt{\sum_{k=1}^{19}(angle_{j,k} - \overline{angle}_j)^2}} \tag{2.27}$$

where $Corr_{i,j}$ stands for the Pearson's correlation coefficient between the i^{th} and the j^{th} gestures and $angle_{i,j}$ represents the angle in the k^{th} joint of the i^{th} posture. Moreover, $\overline{angle}_x$ represents the sample average of x^{th} gesture and is defined as:

$$\overline{angle}_i = \frac{1}{19}\sum_{k=1}^{19} angle_{i,k} \tag{2.28}$$

In addition, to show the correlations for all five pairs (with and without dimensionality reduction), we created a five by five matrix in which the color of the element on the i^{th}

row and j^{th} column represents the Pearson's correlation coefficient between the i^{th} and the j^{th} gesture in the joint angle space. This correlation matrix is defined as follows:

$$C = \begin{bmatrix} Corr_{1,1} & \cdots & Corr_{1,j} \\ \vdots & \ddots & \vdots \\ Corr_{i,1} & \cdots & Corr_{i,j} \end{bmatrix} \tag{2.29}$$

2.4.3 Results

The joint angle representation of the hand gestures shown in Fig. 13 are illustrated in Fig. 2.14a. We apply PCA (as explained in the method section) to filter out the most common component for the five gestures. 91.40% of the variance is explained by the first two PCs. The resulting joint angle space representation for the first two PCs is shown in Fig. 14b.

As can be seen in Fig. 2.14, with only considering the first two PCs, power grasp 1 and power grasp 2 have grouped together. The same pattern is observed with precision grasp 1 and precision grasp 2. This was within our expectations since by saving the first two PCs, we are ignoring the smaller differences and paying attention to the commonalities.

Fig. 2.15 shows the correlation values for the two cases studied on Fig. 2.14 colored boxes in the matrix shown in Fig. 2.15(a) represent the correlation coefficient between the joint angle representation vectors of different gestures. Fig. 2.15(b) represents the same measure for the case with each gesture only being represented by their first two PCs. Comparing Fig. 2.15(a) and 2.15(b), we can make three very important observations.

First, in Fig. 2.15(b), gestures are clustered into three main groups; namely, Power grasp, Precision grasp, and the Claw. These clusters are represented as red squares in Fig. 2.15(b). This shows that by keeping only the first two PCs, the intra-group correlation values have increased i.e. intra-group dissimilarities have decreased. Furthermore, all group members have lost their distinctions in the reduced dimension space to some extent.

Second, the correlation between gestures in power grasp and gestures in precision grasp groups (inter-class similarity) have become smaller and nearly converged to the same value for any pair of gestures from these two groups. This is illustrated as the dark and light blue lines on the intersections of the power grasp and precision grasp clusters in Fig 2.15(b). This is significant because in the reduced dimension space, dissimilarity values between any of the power grasp gestures with any of the precision grasp gestures are almost the same. This means that distinguishing different gestures from each other is much more difficult in the reduced dimension space.

Third, the Claw gesture is much more closely correlated to other gestures in the reduced dimensional space. This is mostly observed between the correlation values of the Claw and power grasp 2, and also between the Claw and precision grasp 2.

This, again, makes an accurate distinction between the different gestures more challenging. This was observed even though the Claw gesture is a non-practical gesture, which is unlikely to be used in day to day activities.

2.5 Chapter conclusions and future work

In this chapter, we took three different approaches to address the question: “Should anthropomorphic systems be redundant?”

In the first section, we presented the classical approach to muscle redundancy for joint torque and endpoint force production in tendon-driven systems. The notion of muscle redundancy holds that there are many ways in which tendon-driven anthropomorphic systems can activate muscles to generate the desired net torques at each joint [19, 15]. However, we underscore that tendon-driven systems are, in fact, *over-determined* from the perspective of *tendon excursions* [148, 43, 150]. That is, the lengths and velocities of all muscles crossing a joint, or set of joints, are determined by the rotations at those joints. From a mechanical perspective, this means that a given limb movement defines a unique set of tendon excursions and velocities. Muscles that shorten during the movement can, in principle, go slack (but then they do not contribute to torque production). However, muscles that lengthen during the movement must do so as specified by the joint rotations. This poses a practical problem in the case where motors are not backdrivable or muscles have stretch reflexes: any muscle that does not lengthen appropriately will disrupt the movement. Therefore, the controller (be it neural or engineered) seeking to produce smooth and accurate movement in a tendon-driven system is not necessarily confronted with a redundant system, as is typically assumed. Rather, it must excite muscles to produce the necessary time history of joint torques *while* allowing muscles to lengthen in the precise way needed. This perspective is not new. Sherrington emphasized the importance of inhibition as a central requirement for the production of movement over 100 years ago [139, 140]. We therefore propose that it is critical for researchers today to pursue a neo-Sherringtonian research direction to understand the robotic and neural control of movement.

The fact that moving smoothly and accurately is neither a redundant, simple, nor a forgiving control problem for tendon-driven systems poses several critical research directions. For example, why does such behavior take years to perfect during typical development in humans (and still not fully available in robot), and why is it so susceptible to developmental and neurological conditions? We propose that the over-determined nature of muscle excursions makes the problem of producing smooth and accurate movements unforgiving to even small errors in development and neurological conditions, which requires further study [148, 43, 150].

In addition, there is emerging evidence that cardinal features of healthy force and movement variability (which are often considered to have cortical origins) can arise naturally as a consequence of the neural control of afferented muscles (i.e., where regulating reflex gains is critical) [73, 56, 104]. This opens new research directions to begin to explain, from a purely spinal and peripheral perspective, the clinical presentation of at least some types of tremor in neurological conditions.

The second approach described some aspects of Feasibility Theory, which helps us understand how the anatomy of the plant, and the mechanical constraints that define the task, determine the dimensionality and structure of its feasible activation set (i.e., the family of all feasible commands that can accomplish the task). Such feasible activation sets are well-structured, low-dimensional subspaces embedded in the high-dimensional space of muscle activations. Thus, future research should focus on how the controllers of anthropomorphic systems can explore, identify, exploit, and remember those feasible activation sets. After all, the most any neural or engineered controller can do is explore and exploit the capabilities of the tendon-driven system as a whole [148].

Another important aspect of Feasibility Theory is that the number of independently-controllable muscles also determines the number of independent task constraints that tendon-driven systems can satisfy [148, 54]. Thus, adding and having more (appropriately placed) muscles may, in fact, be the critical enabler of ecological (i.e., real-world) function. That is, more muscles enable performing more complex tasks—where complexity is taken to mean the need to meet more task constraints simultaneously. Thus, it is important to investigate how ecological tasks necessitate having more muscles than are apparently necessary when studying "simpler" experimental tasks [78]. This implies that failure to control all muscles independently (as is common in, say, stroke) will reduce functional capabilities because independent muscle control is necessary to meet the multiple requirements of ecological tasks. Thus, “redundant” systems with many muscles (or control DOFs in general) are functionally desirable.

Importantly, these findings also motivate further research on the advantages and disadvantages of muscle synergies (where several muscles are activated in a correlated manner, effectively reducing the number of independent control DOFs). Given that implementing muscle synergies can be an effective way to control robots [133]—and assess limb movement [145]—exploring the relationship between the number of independent control DOFs and functional versatility requires further study [51, 10, 71, 151].

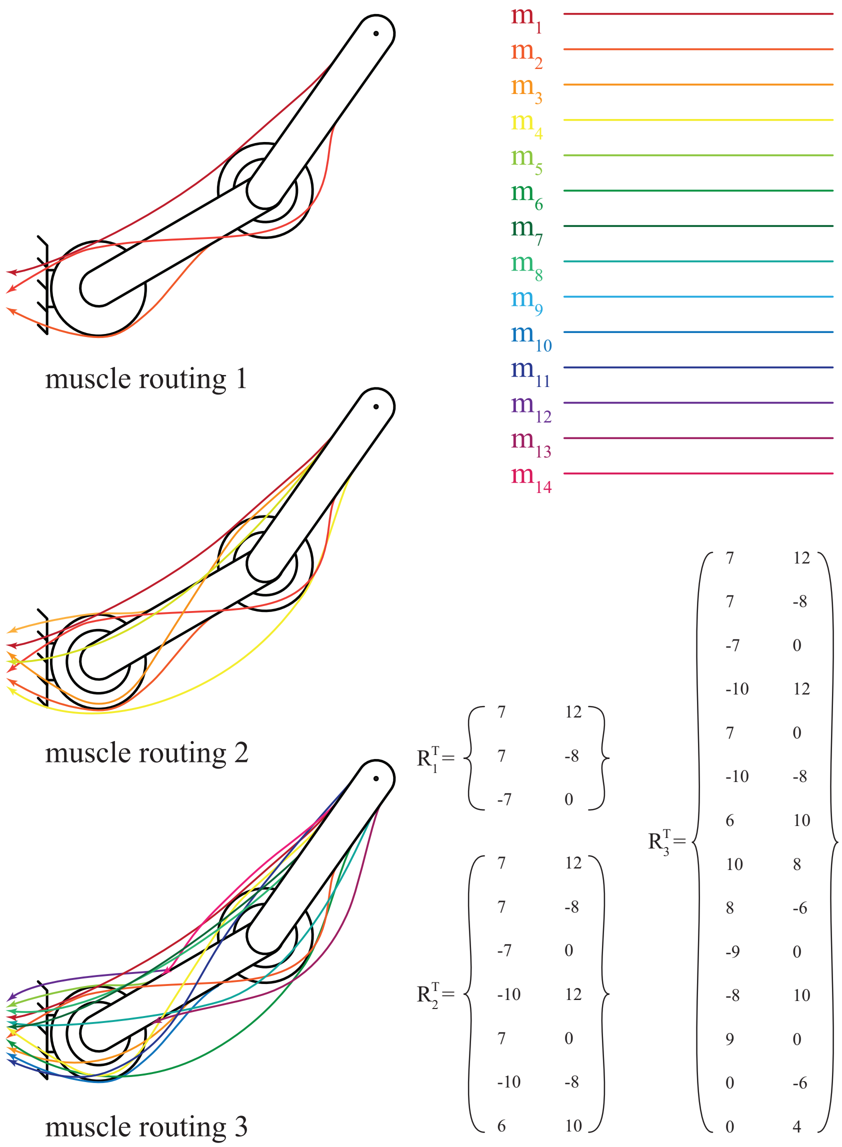
The second section also approached the classical problem of muscle redundancy from the perspective of multi-objective optimization. That is, tackling simultaneous and independent functional goals. As an example, we explored three: maximal end-point force, maximal joint angular velocities, and the Phenotypical cost of having additional muscles. We showed how an anthropomorphic system can adapt to consider all three goals to arrive at a most desirable number of muscles (yet sub-optimal with respect to individual cost functions). Moreover, this desirable number of muscles is a function of the relative weighing across goals. We find that more muscles allow the limb to be better at multiple goals. Moreover, this study underscored how a change in environment, goals, and life habits may, over time, naturally change the number and/or routing of muscles in a given anthropomorphic system, and vice versa.

The third and last section highlights our final approach to the question: “Should anthropomorphic systems be redundant?” We first showed how, in agreement with [134], more than 90% of the variance in different hand gestures studied here can be explained with only two principal components. However, disregarding the remaining PCs will naturally make it more difficult to distinguish and/or implement each gesture. This example highlights a little-appreciated—in our opinion—consequence in dimensionality reduction: that it will make it very difficult to distinguish similar hand gestures with different functional roles as per well-known grasp taxonomies [28]. For example, a power grasp with the thumb abducted serves to oppose the fingertips. If the thumb is slightly adducted, it can press against the side of the fingers to roll an object with high precision (see Figure 3 in [28]). Thus, although the higher PCs explain ever-decreasing percentages of variance, they nevertheless have important functional consequences. This is in agreement with recent findings in the field of soft robotics. Such studies show how small amounts of passive deformation, provided by non-stiff materials, can significantly increase the functional capabilities, robustness, and versatility of under-actuated hands [10, 21, 12]. Therefore, finding that some PCs that explain relatively little variance does not necessarily mean that the system has unnecessary DOFs or is functionally redundant.

MATLAB toolbox All the simulations demonstrated in this chapter were performed using a custom Neuromechanics toolbox written in MATLAB. This toolbox is available at github.com/marjanin/Neuromechanics-Toolbox and at ValeroLab.org.

Acknowledgements This study was supported by the National Institute of Arthritis and Musculoskeletal and Skin Diseases of the National Institute of Health (NIH) under award numbers R01 AR-050520 and R01 AR-052345 to FVC, and University of Southern California Graduate School's Provost Fellowship to AM. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

We acknowledge Dr. Christopher Laine and Amir Ahmadi for their helpful comments on this manuscript. We thank the University of Southern California for facilities provided during the course BME/BKN 504 and Antara Dandekar and Monica Guerrero for their help with the conceptualization of Section II.



$$R_1^T = \left\{ \begin{array}{cc} 7 & 12 \\ 7 & -8 \\ -7 & 0 \end{array} \right\}$$

$$R_2^T = \left\{ \begin{array}{cc} 7 & 12 \\ 7 & -8 \\ -7 & 0 \\ -10 & 12 \\ 7 & 0 \\ -10 & -8 \\ 6 & 10 \end{array} \right\}$$

$$R_3^T = \left\{ \begin{array}{cc} 7 & 12 \\ 7 & -8 \\ -7 & 0 \\ -10 & 12 \\ 7 & 0 \\ -10 & -8 \\ 6 & 10 \\ 10 & 8 \\ 8 & -6 \\ -9 & 0 \\ -8 & 10 \\ 9 & 0 \\ 0 & -6 \\ 0 & 4 \end{array} \right\}$$

Figure 2.8: Muscle routings and moment arm matrices used in the simulations.

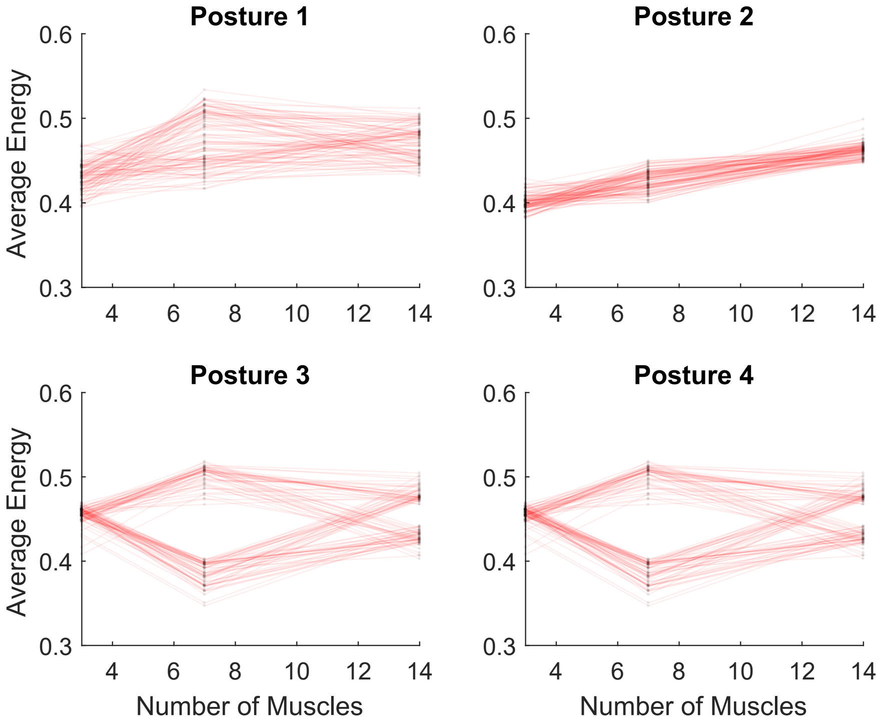


Figure 2.9: Average Energy plots (Monte Carlo analysis).

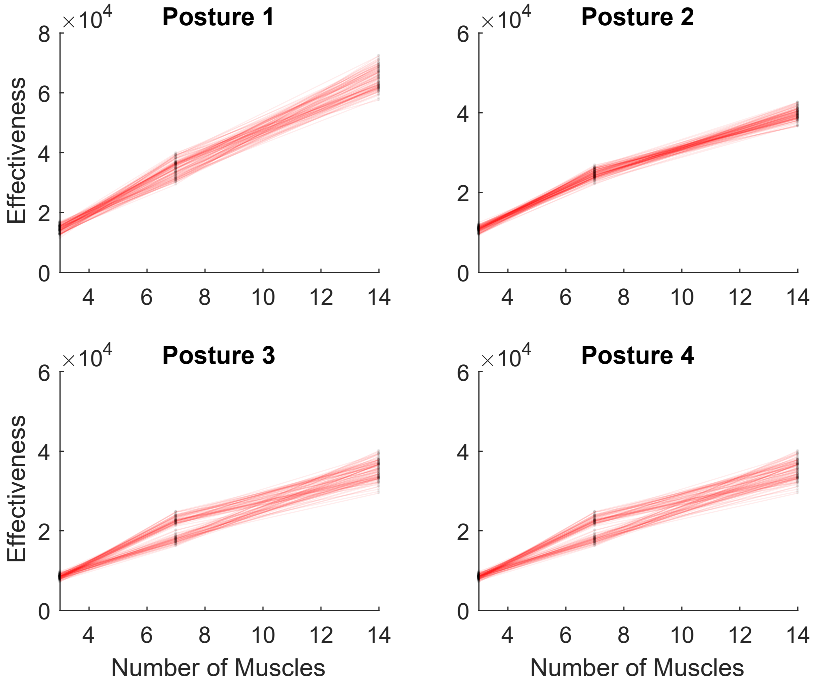


Figure 2.10: Effectiveness plots (Monte Carlo analysis).

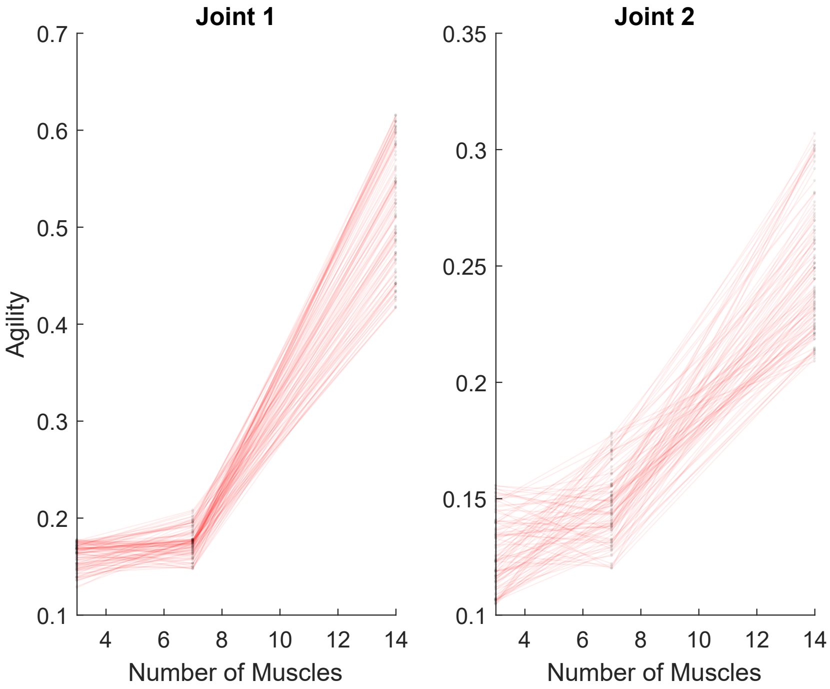


Figure 2.11: Agility plots for each of the two joints (Monte Carlo analysis).

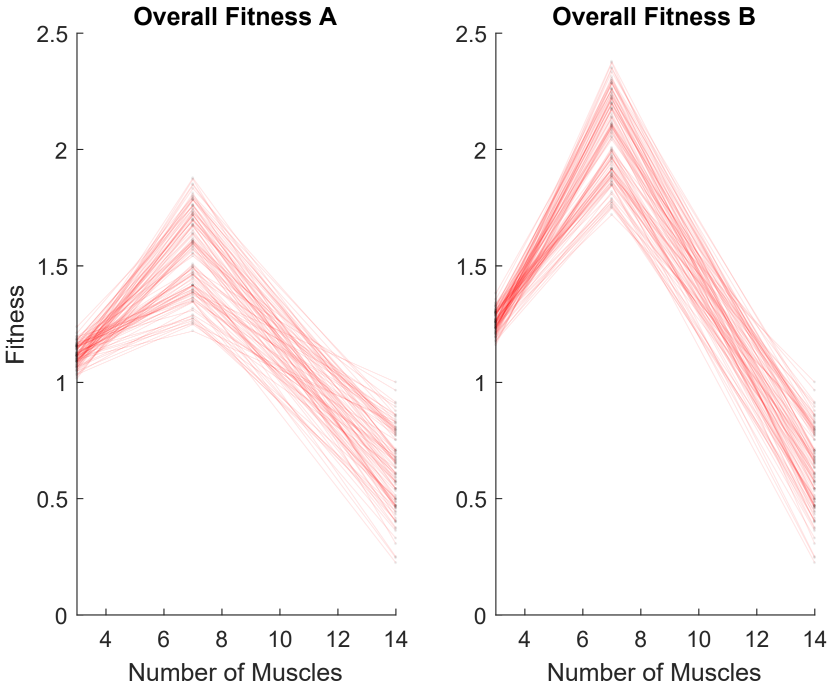


Figure 2.12: Overall Fitness A and B as a function of the number of muscles (Monte Carlo analysis). Note that Phenotypical costs are quadratic (A) or cubic (B) functions that dominate for larger numbers of muscles.

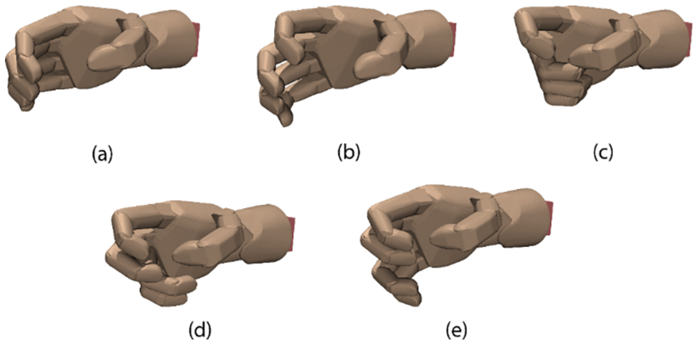


Figure 2.13: 3D model of the five different hand gestures studied in this section. (a) Power grip 1. (b) Power grip 2. (c) Precision grip 1. (d) Precision grip 2. (e) The Claw gesture.

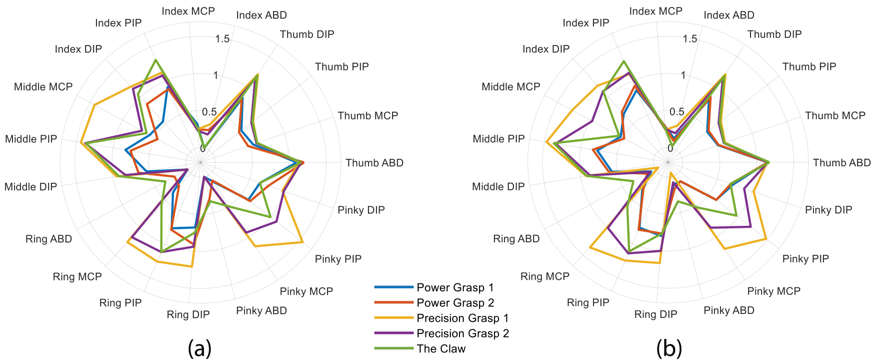


Figure 2.14: Spider plot representations of the 19 joint angles (a) Without dimensionality reduction. (b) With only the first two PCs.

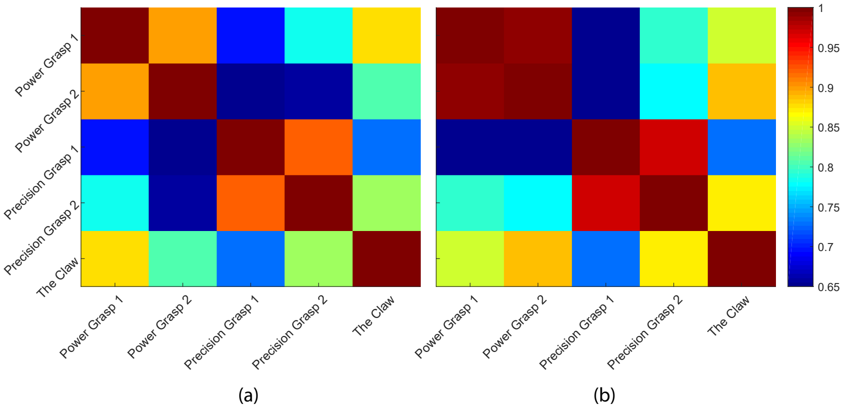


Figure 2.15: The correlation coefficients for each of the gesture pairs. (a) Without dimensionality reduction. (b) With only the first two PCs.

Chapter 3

Autonomous Functional Movements in a Tendon-Driven Limb via Limited Experience

Ali Marjaninejad[1], Darío Urbina-Meléndez[1], Brian A. Cohn[2], Valero-Cuevas FJ[1,2,3]

[1]Department of Biomedical Engineering, University of Southern California, Los Angeles, CA

[2]Department of Computer Science, University of Southern California, Los Angeles, CA

[3]Division of Biokinesiology and Physical Therapy, University of Southern California, Los Angeles, CA

3.1 Chapter summary

Robots will become ubiquitously useful only when they can use few attempts to teach themselves to perform different tasks, even with complex bodies and in dynamical environments. Vertebrates, in fact, use sparse trial-and-error to learn multiple tasks despite their intricate tendon-driven anatomies—which are particularly hard to control because they are simultaneously nonlinear, under-determined, and over-determined. We demonstrate—for the first time in simulation and hardware—how a model-free, open-loop approach allows few-shot autonomous learning to produce effective movements in a 3-tendon 2-joint limb. We use a short period of motor babbling (to create an initial inverse map) followed by building functional habits by reinforcing high-reward behavior and refinements of the inverse map in a movement's neighborhood. This biologically-plausible algorithm, which we call G2P (General-to-Particular), can potentially enable quick, robust and versatile adaptation in robots as well as shed light on the foundations of the enviable functional versatility of organisms.

3.1.0.1 Author Contribution

A.M. led the development of the G2P algorithm, D.U.-M. led the construction of the robotic limb, B.C. led the data acquisition and analysis. F.V-C. provided general direction for the project. All persons designated as authors qualify for authorship, and all those who qualify for authorship are listed. All authors also contributed to the conception and design of the work, and writing of the manuscript.

3.2 Introduction

Today's successful control algorithms for robots often require a combination of accurate models of the physical system, task, and/or the environment or expert demonstration of the task; as well as expert knowledge to adjust parameters or extensive interactions with the environment [81, 3, 65, 109, 113, 83, 93, 36, 22, 9, 67, 58]. Even then, many rely heavily on error corrections via real-time state observation or error feedback [65, 109, 36, 22, 143, 84, 33, 69, 131, 123, 47, 25]. Moreover, some prefer to focus on simulated behavior of simplified systems and environments or limit the physical system to simple scenarios (e.g. only kinematical control) [93, 33, 69, 128, 117, 24, 7, 105, 86, 127, 136]. Although advances in machine learning demonstrate that RL agents can achieve human-like performance in complicated tasks (e.g., video games), or can find optimal strategies for mechanical tasks using evolutionary algorithms, those studies are limited to computer simulations due to the numerous attempts needed for the algorithm to converge[100, 132, 153]. In addition, some researchers seek to apply biologically-plausible principles from anatomy and neuroscience to develop versatile robots and learning strategies [65, 83, 93, 131, 47, 7, 105, 97, 118, 82]. In particular, there is need to develop feed-forward, model-free approaches that learn using limited interactions with the environment (i.e., "few-shot" learning [129]), which could imbue robots with the enviable versatility, adaptability, resilience, and speed of vertebrates during everyday tasks [109, 9, 38, 6, 50]. This work presents a combination of hardware and software advances (in contrast to much current work in robot learning which is done in simulations only) that demonstrate how a model-free, open-loop approach allows few-shot autonomous learning to produce effective movements in a 3-tendon 2-joint limb. Moreover, our approach (Figures 3.3.1 and 3.3.1) is biologically-plausible at two levels: First, we use motor babbling—as do young vertebrates [29, 1]—to learn the general capabilities of the physical systems (also called "plant" in control theory); followed by reinforcement of high-reward behavior and refinements that are particular to the task (i.e., General-to-Particular, or G2P). And second, we use tendons to generate torque at each joints (Figure 3.3.1 and Supplementary Fig. 3.3.1) to replicate the general problem biological nervous systems face when controlling limbs [148] (which makes for a simultaneously over-and under-determined control problem, see Methods) that may lead to a class of robots with unique advantages in design, versatility, and performance [92]. This work also contributes to computational neuroscience by providing a biologically- and developmentally-tenable learning strategy for anatomically-plausible limbs (Supplementary Discussion).

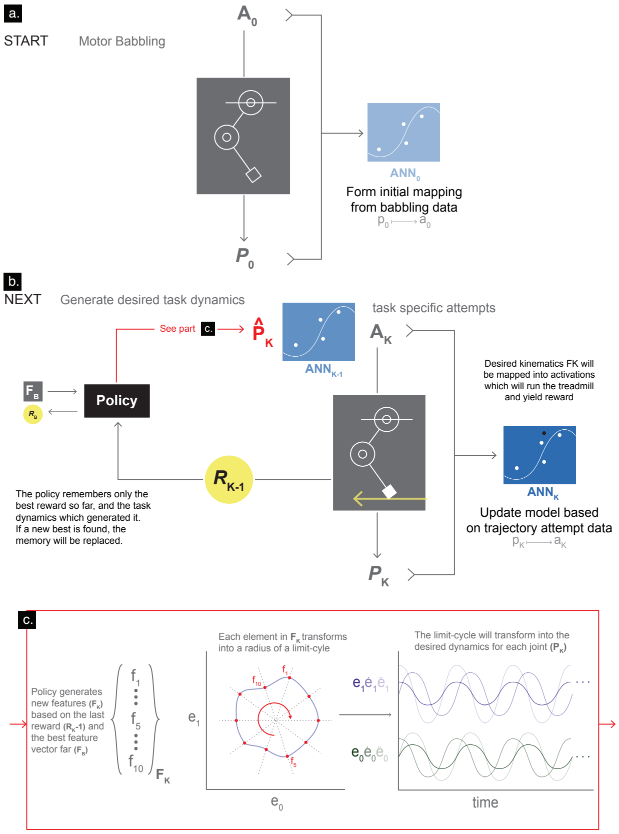
3.3 Results

We show that the G2P algorithm can autonomously learn to propel a treadmill (while supported by a carriage) without closed-loop error sensing, or an explicit model of the dynamics of the tendon-driven limb or the environment (e.g., limb inertia, contact dynamics, or expected reward). We also show that execution of multiple attempts can itself lead to improvement in performance on account of a refined inverse map in the neighborhood of the movement. Such cost-agnostic improvements serve as a proof-of-principle of a biologically-tenable mechanism that benefits from familiarity with the task, rather than teleological optimization, or even error-driven corrections.

3.3.1 Results for cyclical movements to propel the treadmill

A given run begins with a 5-minute motor babbling session where the time-history of a pseudo-random control sequence (a 3-D time-varying vector of step changes of current to each motor) is fed to the limb while its kinematics (joint angles, angular velocities and angular accelerations) are measured by encoders at each joint (Fig. 3.3.1 shows an overview of G2P). An Artificial Neural Network (ANN) then uses these motor babbling data to create an initial inverse map from 6-dimensional kinematics to 3-dimensional control sequence. A movement to propel the treadmill is parameterized by a closed orbit in 2-dimensional joint-angle space that interpolates between the "feature vector" of 10 evenly-distributed points (Fig. 3.3.1c). For a given cycle duration of 1s, this defines the 6-dimensional limb kinematics: joint angles, angular velocities and angular accelerations for each of the two joints; see Methods for details). Next, 20 replicates of these kinematics are fed through the initial inverse map (lower-level control) which produces 20 cycles of a control sequence (Fig. 3.3.1c). Those control sequences are delivered to the robotic limb to produce 20 cycles. The reward for that attempt is a scalar value representing the distance the treadmill was propelled backward, in millimeters (mm), as in forward locomotion. Reward for each attempt is provided to the system in a discrete way (only after the attempt—20 cycles—is over).

A sequence of attempts (Fig. 3.3.1) within each run of the G2P algorithm (Fig. 3.3.1) uses the initial inverse map to start the exploration phase: the ten free parameters of the feature vector are changed at random and the resulting dynamics are sent to refine the inverse map. The resulting control sequence is fed to the motors to produce limb movement until the treadmill reward crosses a threshold of performance set to 64 mm (empirically selected to lead to clearly observable propulsion). Thereafter, the exploitation phase of G2P begins: we use policy-based Reinforcement Learning (RL) with stochastic policy search in which the feature vector is sampled from a 10-dimensional Multivariate Gaussian distribution. The mean vector of this Gaussian distribution is the best feature vector (i.e., that yielded the highest reward so far), and its standard deviation (SD) values shrink as the reward increases (see Methods). Feature vectors sampled from this Gaussian are used in subsequent attempts. Those that produce higher reward serve as the new best feature vector (see Methods for more detail). This process resembles an evolutionary algorithm and is similar to cross-entropy optimization method with the distinction than here we just use one candidate solution (as opposed to a population of solutions) and the SD is a function of the reward (as opposed to SD of the sub-population with highest rewards). Each time a control sequence is applied (in either the exploration or exploitation phase), the resulting kinematics are recorded, appended to the babbling data and any prior attempts, and included in the next refinement of the inverse map (Fig.



Form initial mapping from babbling data

Generate desired task dynamics

Desired kinematics FK will be mapped into activations which will run the treadmill and yield reward

Update model based on trajectory attempt data

The policy remembers only the best reward so far, and the task dynamics which generated it. If a new best is found, the memory will be replaced.

Policy generates new features (F_K) based on the last reward (R_K-1) and the best feature vector far (F_B)

Each element in F_K transforms into a radius of a limit-cyle

The limit-cycle will transform into the desired dynamics for each joint (P_K)

Figure 3.1: The G2P algorithm - Every run of the algorithm begins with (a) time-varying babbling control sequences (activations A_0 that run through the electric motors) that generate five minutes of random motor babbling (P_0). These input-output data are used to create an inverse (output-input) map ANN_0 from limb kinematics to control sequences. (b) Reinforcement learning begins by varying the ten free parameters of the feature vector (F_K) defining a cyclical movement. These movements can, in principle, propel the treadmill. ANN_0 maps each candidate desired kinematics (($\hat{P}_K$)) into activation sequences (A_K) which will run the treadmill (P_K being resulting kinematics) and yield reward (R_K). An attempt (K being the attempt counter) is when an activation sequence is repeated twenty times and used to produce twenty steps worth of kinematic data. These kinematic data are further processed and concatenated with all prior data to refine the inverse map into ANN_K. The total treadmill propulsion, if any, is the reward for that attempt. The policy remembers only the best reward so far, and the task dynamics which generated it. If a new best is found, the memory will be replaced. (c) If the new reward exceeds the best so far (R_B), the policy updates its memory of the best feature vector so far (F_B) and continues its search in the increasingly smaller neighborhood of that feature vector by sending the resulting kinematics to the ANN. But note that data from all attempts (whether they improve on the best so far or not) are used to refine the inverse map. Fig. 3.3.1 describes data processing for each run.

3.3.1b). That is, every interaction with the physical system is used in the next attempted refinement of the inverse map. This is analogous to trial-to-trial experiential adaptation during biological motor learning40.

Fig. 3.3.1a shows the reward for each sequential attempt for 15 independent runs labeled A—O. These color-coded stair-step lines show the best reward achieved thus far. Our system was able to cross the exploration-exploitation threshold in a median of 24 attempts, and the subsequent exploitation phase showed median reward improvement of 45.5mm with a final reward median of 188mm (best run performance was 426.9mm). Simulation results for the corresponding test are shown on Supplementary Fig. 3.3.1.

Fig. 3.3.1b shows that the system is able to learn families of related solutions (i.e., a motor habit), and that—for each such family—high rewards can be achieved with both high and low power consumption. This shows that energy minimization is not an emergent property in this biologically-plausible system or learning strategy. However, if desired, an energy optimization term could be appended to the reward to yield this property.

3.3.2 Results for free cyclical movements in air

The utility of familiarity with a task to produce incremental improvements (by increasing the precision of inverse map) cannot be directly interpreted from the results in Fig. 3.3.1. This is because the reinforcement learning algorithm might, by itself, find a feature vector that yields high reward even with an imprecise inverse map. However, in many applications, such as tracking a desired trajectory (a form of imitation), precision

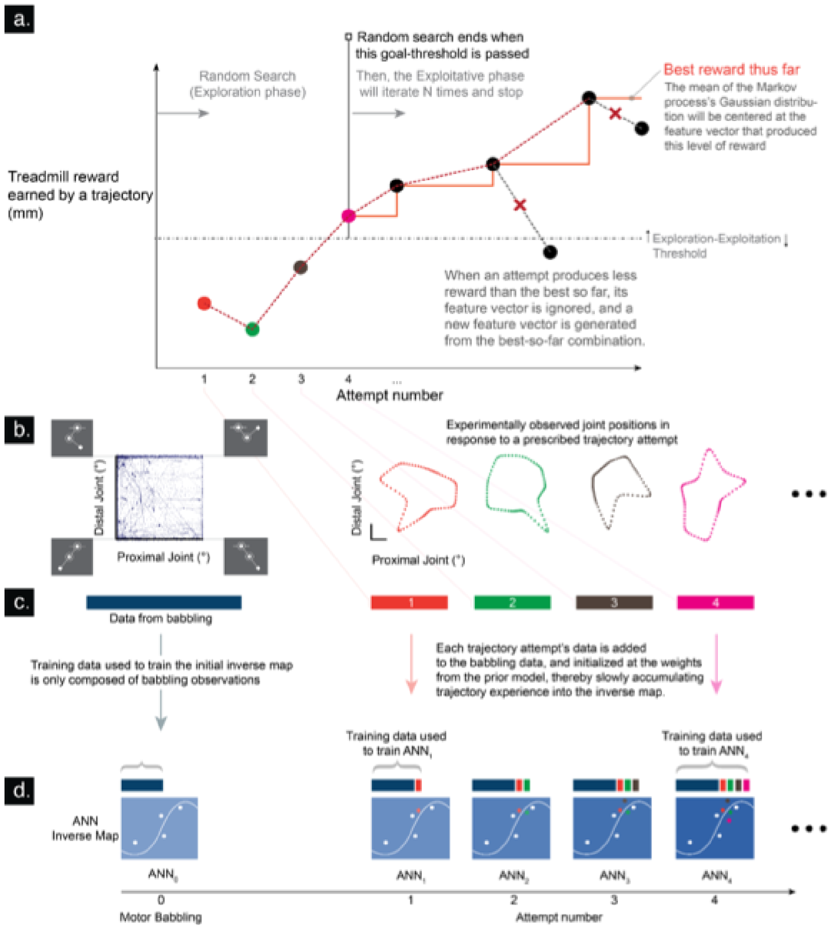

a.
Random search ends when this goal-threshold is passed
Random Search (Exploration phase)
Then, the Exploitative phase will iterate N times and stop
Best reward thus far
The mean of the Markov process's Gaussian distribution will be centered at the feature vector that produced this level of reward
Treadmill reward earned by a trajectory (mm)
Exploration-Exploitation Threshold
When an attempt produces less reward than the best so far, its feature vector is ignored, and a new feature vector is generated from the best-so-far combination.
1
2
3
4
...
Attempt number
b.
Experimentally observed joint positions in response to a prescribed trajectory attempt
Distal Joint (°)
Proximal Joint (°)
Distal Joint (°)
Proximal Joint (°)
c.
Data from babbling
1
2
3
4
Training data used to train the initial inverse map is only composed of babbling observations
Each trajectory attempt's data is added to the babbling data, and initialized at the weights from the prior model, thereby slowly accumulating trajectory experience into the inverse map.
d.
Training data used to train ANN1
Training data used to train ANN4
ANN Inverse Map
ANN0
ANN1
ANN2
ANN3
ANN4
0
Motor Babbling
1
2
3
4
Attempt number

Figure 3.2: A run of the G2P algorithm in detail for the reward-driven treadmill task - (a) Evolution of reward across the exploration and exploitation phases. The exploration phase begins by using the initial inverse map ANN0 (Fig. 3.3.1) to attempt to produce the cyclical movement defined by the first feature vector selected from a random uniform distribution. The predicted control sequence is applied to the motors to produce twenty cycles of movement that yield a particular treadmill reward (orange dot) and continues to be changed until a feature vector is found that yields a reward above the exploration-exploitation threshold (dotted line). It then transitions to the exploitation phase where the feature vectors of the subsequent 15 attempts are sampled from a 10-dimensional Gaussian distribution centered on the best feature vector so far. Motor babbling and sequential task-specific refinements of the inverse map: (b) Distribution of the proximal and distal joint data from motor babbling (enlarged in Figure 3.4.2) and subsequent attempts (color coded). (c) Babbling data (shown schematically as a blue bar) were used to generate the initial inverse map (ANN_0), and (d) concatenated with data from each attempt to continually refine the inverse map (ANN_1, ANN_2, . . .).

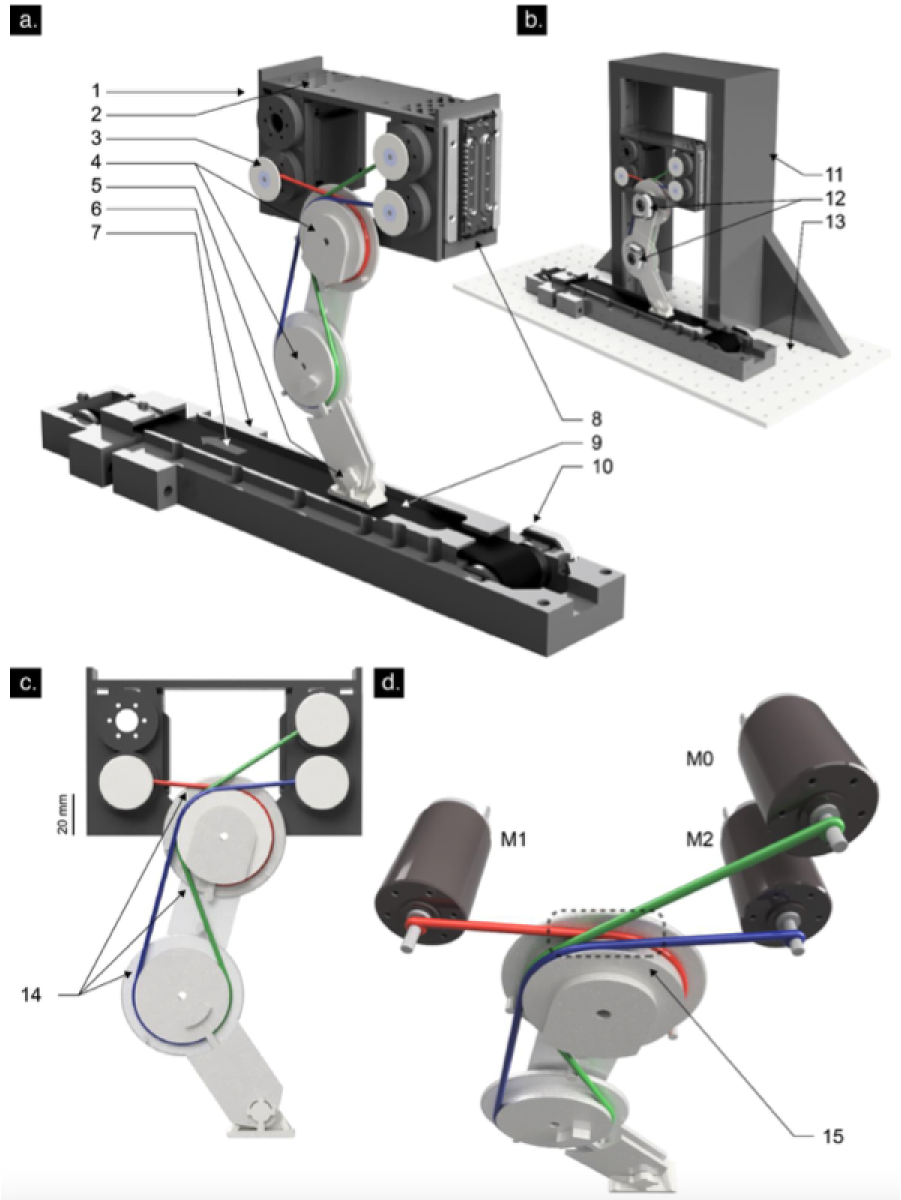


Figure 3.3: Planar robotic tendon-driven limb - (a) General overview of the physical system 1. Motor-joint carriage 2. Motor ventilation 3. Shaft collars 4. Joints (proximal and distal) 5. Passive hinged foot. 6. Treadmill 7. Direction of positive reward 8. Linear bearings on carriage (locked at a particular height during testing) 9. Treadmill belt 10. Treadmill drum encoder. (b) Fully supported system 11. Frame 12. Absolute encoders on proximal and distal joints 13. Ground. (c) Tendon routing 14. Three tendons driven by motors M0, M1 and M2. (d) System actuation. Motor M1 drives only the proximal joint ccw, while M0 and M2 drive both joints (M0 drives the proximal joint cw, and the distal joint ccw, while M2 drives both joints cw). 15. Tendon channel.

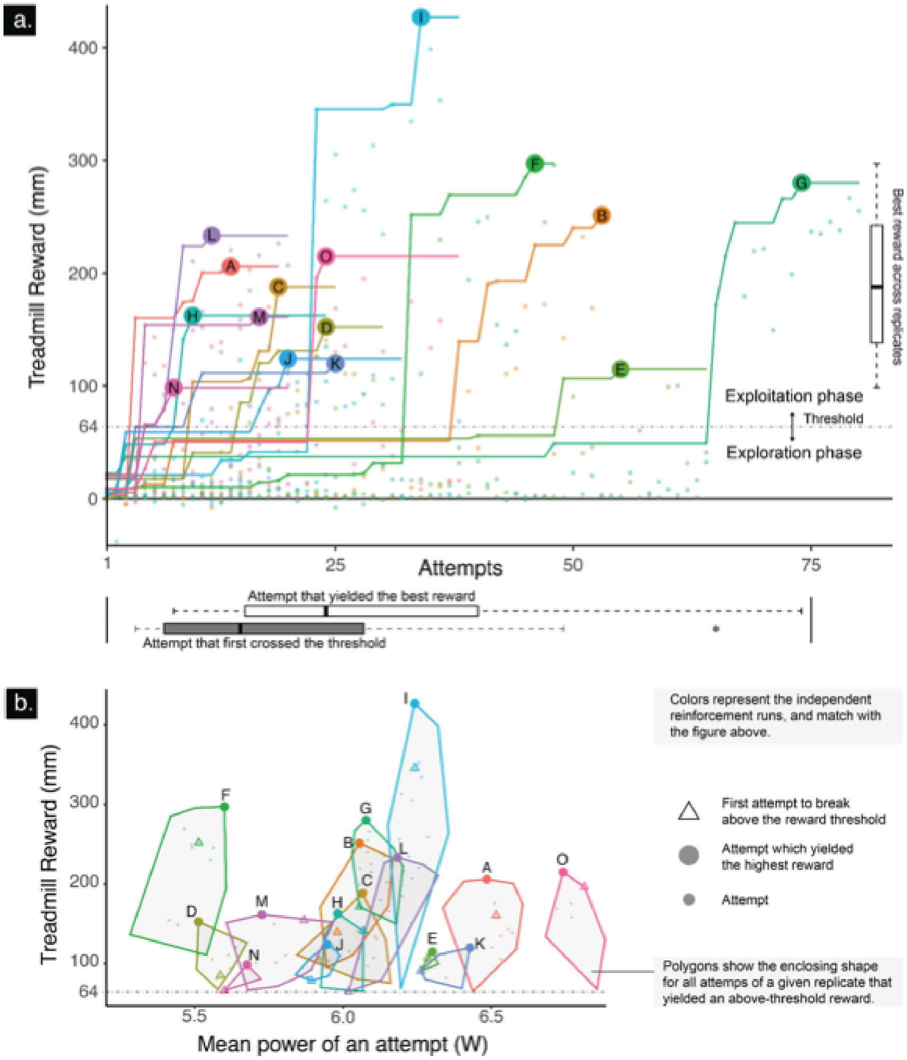


Figure 3.4: The treadmill task results - (a) Treadmill reward accrued in each of fifteen independent runs, labeled A—O: All runs crossed the exploration-exploitation threshold of 64 mm of treadmill propulsion (median of exploration attempts: 15). All runs showed improvement, where the median number of attempts needed to reach the best reward of each run was 24. (b) Reward vs. energy consumption (Mean power of an attempt): We plot all attempts from runs which garnered a reward above the exploration-exploitation threshold on the reward vs. energy consumption plane. We can then find the convex hull representing them as a family of similar solutions, or a motor habit. For each polygon, the peak reward (large dot) and the reward from the first attempt to cross the threshold (triangle) are shown. We detect no right-to-left trend indicating that energy consumption was spontaneously reduced as performance improved. Conversely, higher reward did not always require higher energy consumption even though more external work was being done to propel the treadmill the furthest.

of this inverse map is crucial. We, therefore, performed two trajectory-tracking tasks in air (with no explicit reward or real-time feedback) to evaluate the performance of G2P in refining the inverse map during task-specific explorations for a given cyclical trajectory as well as the generalizability of these refinements on unseen cyclical trajectories.

(A) **Free cyclical movement in air for a single trajectory**

The limb was suspended ‘in the air’ without making contact with the treadmill while, as before, the initial inverse map was extracted from five minutes of motor babbling data. For each run, this initial inverse map (ANN_0) was incrementally refined with data from each of five attempts, regardless of its tracking error over the course of the attempt. Figures 5(a-i) show reduction of the Mean Square Error (MSE) with respect to the attempt number for one sample run. Figures (A)(a-ii,-iv). show the time history of actual achieved vs. desired joint angles for those same five attempts (see Supplementary Fig. 3.3.1a-b for the simulation result of the corresponding test). Supplementary Fig. 3.3.1 also shows the boxplots of the number of iterations for babbling and the following 4 refinements over 50 replicates using data recorded from the physical system during this task.

(B) **Generalizability of learned free cyclical movements in air**

Although we have demonstrated how repeated exposure to a same task improves performance of that task (A. above and (A)(a)), this does not speak to the generalization of a given inverse map to the execution of other unseen trajectories. Here, we followed motor babbling with serial refinements over thirty randomly selected

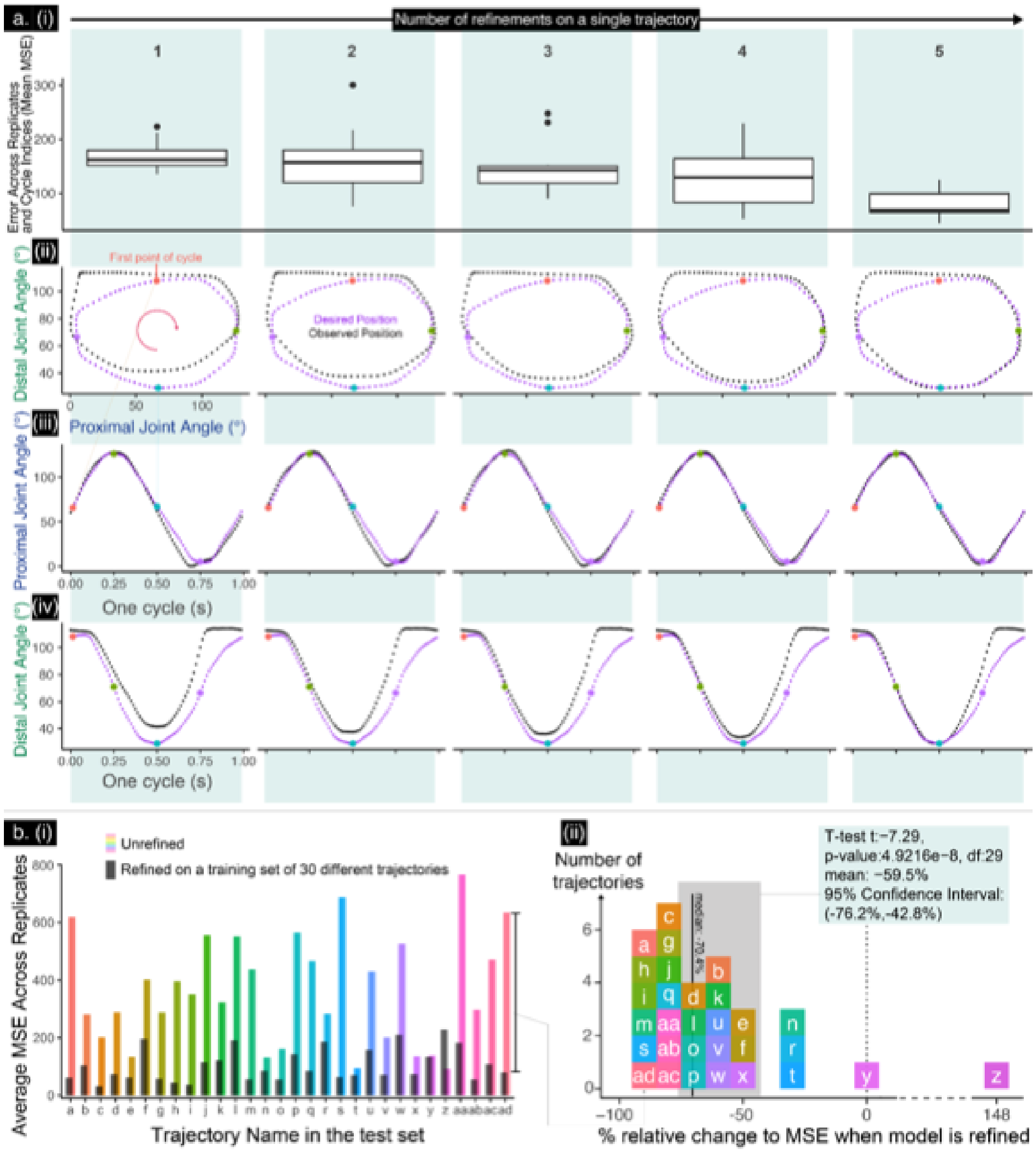


Figure 3.5: A run of the G2P algorithm in detail for the tracking of free cyclical movements - (a) Improvements in performance resulting from 5 attempts at producing a target cyclical movement defined by a given feature vector: (a-i) Boxplots of Mean Square Error (MSE). (a-ii to a-iv). Desired vs. actual joint kinematics. (b) Test of generalization of refined model over unseen trajectories a, b,. . . , ad (see text): (b-i) MSE of the 30 test trajectories executed using either an unrefined inverse map (only babble-trained, color bars) or refined inverse map (sequentially over 30 other training trajectories, gray bars). (b-ii) Histogram of percent difference in MSE for the results in (b-i) for each of the 30 unseen test trajectories.

trajectories (features sampled from a uniform distribution within 0.2-0.8 range). The trained inverse map was then "fixed" and evaluated for its MSE accuracy on 30 additional unseen random (same random distribution) trajectories (the test set) without further refinement. Fig.(A)(b-i,-ii) show that this refined inverse map performed better on the test set. This strongly suggests that refining a map with specific examples improves performance on a variety of test tasks and does not over-fit to its training set. As such, the refined map captures well the complex mechanics of the tendon-driven double-pendulum limb to produce dynamical cyclical movements. This is very important since it means G2P can learn from every experience and generalize it to similar tasks (see Supplementary Fig. 3.3.1c for the simulation result of the corresponding test). The fact that we stack all data (babbling and every new experience) to refine the ANN enables the system to improve performance for other related tasks without forgetting the old ones (see Methods).

3.3.3 Robustness to Perturbation

In a variant of test A. above (after babbling and 10 refinement attempts), we struck the limb with a metal rod once the system was moving at steady state. This blunt perturbation pushed the limb away from its cyclical movement, but then the system returns to its steady state behavior after $\sim$ 1 cycle (see Supplementary Video 2). Poincaré return maps and stability analysis for these perturbation tests are available in the supplementary information (see Supplementary Figures (A) and 3.4.2).

3.3.4 Point-to-point and more complex non-cyclical movements

For the point-to-point tests, the system starts at an initial posture and then performs ramp-and-hold transitions to each of 5 different positions in the joint angles space. For the complex, non-periodic task, the system is instructed to follow a non-periodic trajectory for each joint. Each of these trajectories consist of smooth and ramp-and-hold movements (both in-phase and out-of-phase) of each joint (although the other joint might be moving). This is particularly challenging because two of the tendons cross both joints, so isolated movement of one joint requires coordination across all tendons. Supplementary Video 2 shows an instance of each of these tests. The system (which operates open-loop) reasonably performed both tasks. Supplementary Figure 7 provides these results. Although the system's performance for arbitrary and more complex movements needs to be further investigated, these results serve as encouraging proof-of-principle that extends the utility of the G2P algorithm beyond cyclical movements—the focus of this first investigation.

3.4 Discussion

The G2P algorithm produced two important results in the context of the challenging task of few-shot learning of feedforward and robust production of a cyclical movement of a tendon-driven system. This brings novel possibilities to robotics in general as it shows that a few-shot approach to autonomous learning can lead to effective and generalizable control of complex limbs for movements and, by extension, a new generation of biologically-plausible robots for locomotion, manipulation, swimming and flight. Given its biologically-tenable features, G2P can ultimately also enable the control of neuromorphic systems (e.g., [95]) to help explain the versatility of neuromuscular systems.

3.4.1 How does G2P relate to the field?

The G2P algorithm's main contribution is that it combines developmentally- and biologically-plausible approaches in both hardware and software to autonomously learn to create functional habits that produce effective feedforward behavior—where familiarity reinforces habits without claim to uniqueness nor global optimality. Moreover, it does so, based on a data-driven approach that uses few-shots (i.e., limited experience) seeded by motor babbling. Importantly, it does so in the physical world for a biologically-plausible tendon-driven limb for complex dynamical tasks with and without intermittent contact, and not just in simulation. We now discuss how this novel integrative approach compares and contrasts with other work in machine learning, reinforcement learning and control theory.

We used a model-free approach because precise prior knowledge of the system and the environment is not usually available for dynamical tasks in the physical world [109, 36, 9, 38, 6]. This is also the case for systems that rely on experts to manually tune system parameters, select the appropriate hyper-parameters or provide demonstrations of the task [109, 83, 93, 36, 67, 84]. Without such knowledge, the system often needs to execute numerous iterations in the real-world, simulation (real-time or off-line) or both to converge on adequate performance which can make the learning process costly [3, 9, 58, 127, 136, 100, 132, 153]. Therefore, data-driven model-free systems that do not rely on prior knowledge and can learn with minimal experience are needed [9, 38, 6]. A common approach in robotics today is a compromise: use models of a system to first develop controllers in simulation (e.g., [81, 3, 68]), and then deploy them in physical systems (often known as transfer learning).

Although feedback can play an essential role in control; be it biological or engineering, whenever possible, feedforward systems are preferable. It is especially the case where real-time computation is not available, the state cannot be observed reliably, or when delays are large compared to the dynamics of the task [9]. Thus, real-time feedback system can be costly for engineered and biological systems [109]. Alternatively, feedforward control using precise inverse maps can be used to minimize reliance on feedback. Therefore, an efficient system should only utilize feedback when necessary. In fact, this is even the case in biological systems where, for example, movement-related sensory feedback is not necessarily needed for humans to learn to execute a motor skill [50].

Adequate performance in the physical world is a desirable property for any controller, as it demonstrates its robustness to the full set of dynamics and disturbances. Successful control of tendon-driven limbs in real-world physics is a challenging test of learning and control strategies [3, 131, 86, 148, 92]. Roboticists find such anatomies particularly hard to control because they are simultaneously nonlinear, under-determined (many tendon tensions combine to produce few net joint torques), and over-determined (few joint rotations define how many tendons need to be reeled-in/payed-out) [148, 92]. Some have successfully controlled such tendon-driven systems in the real world using feedback control of fingers18 and manipulation [3]. Others have used simulations to produce simple tasks (hopping/point-to-point movements via manual tuning of parameters [93]). Our work is a real-world demonstration of autonomous learning for feedforward control of dynamic cyclical and discrete tasks in a tendon-driven system via few-shot learning and minimal prior knowledge.

3.4.2 Familiarity reinforces habits

Motor babbling creates an initial general map, from which a control sequence for a particular movement is extracted. This initial prediction serves as a "belief" about the relationship between body/environment, and an appropriate control strategy. This prediction is used for the first attempt that, while imperfect, does produce additional sensory data in the neighborhood of a particular task. These data are subsequently leveraged toward refinement of the inverse map, which then leads to an emergent improvement in performance and reinforcement of useful beliefs.

Importantly, the details of a given valid solution are idiosyncratic and determined by the first randomly-found control sequence that crossed the exploration-exploitation threshold of performance (Fig. 3.3.1). Hence all subsequent attempts that produce experience-based refinements are dependent on that seed (much like a Markov process). This solution and its subsequent refinements, in fact, are a family of related solutions can be called a “motor habit” that is adopted and reinforced even though it has no claim to uniqueness nor global optimality [57]. Biologically speaking, vertebrates also exhibit idiosyncrasies in their motor behavior, which is why it is easy to recognize health states, sexual fitness, identify individuals by the details of their individual movement and speech habits, and even tell their styles and moods. A subtle but important distinction is that these emergent motor habits are not necessarily local minima in the traditional sense. They are good enough solutions that were reinforced by familiarity with a particular way of behaving. There is evidence that such multiplicity of sub-optimal, yet useful, set points for the gains in spinal circuitry for discrete and cyclical movements45. Those authors argue that it is evolutionary advantageous for vertebrates to inherit a body that is easy to learn to control by adopting idiosyncratic, yet useful, motor habits created and reinforced by an individual’s own limited experience, without consideration of global optimality [57]. G2P uses a similar learning strategy.

Fig. (A)(a) also demonstrates familiarity as an enabler of learning, where we tested the ability of to produce free cyclical movements in air, without contact with the treadmill—and hence without explicit reward. The performance of a particular free cyclical movement improves simply on the basis of repeated attempts. This represents, essentially, the cementing of a motor habit on the basis of experience in the neighborhood of the particular movement. Fig. 3.4.2 further shows 15 cycles of a particular free movement in the interior of the joint angle space, even though is the most poorly explored region during babbling. Importantly, familiarity with the neighborhood of a task need not lead to overfitting that is only locally useful. Our cross-validation experiments in Fig. (A)(b) show familiarity with one's motion capabilities for some tasks seems to inform the execution of other tasks. Note that the absence of a reward or penalty for particular joint angles allowed the emergent solution to contain a portion where the distal joint is at its limit of range of motion. This, however, need not be detrimental to behavior. For example, human walking often has the knee locked in full extension right before heel strike.

3.4.3 Task reward vs. energetic cost

Studying whether energetic efficiency during locomotion is an emergent property, or must be actively enforced, is a longstanding question in motor control [79, 18, 66]. The results in Fig. 3.3.1b are particularly interesting because they show that energy minimization is not an emergent property in this system [30]. Fig 3.3.1a shows the sequence of attempts from each run. Each family of attempts that perform above the exploration-exploitation threshold (plotted with the polygonal convex hull that includes them; Fig. 3.3.1b) can be narrow or wide from the perspective of energetic cost (horizontal axis), but nowhere do we see a general trend towards energy minimization within families (i.e., none of the convex hulls are shaped diagonally towards the top left). Conversely, one

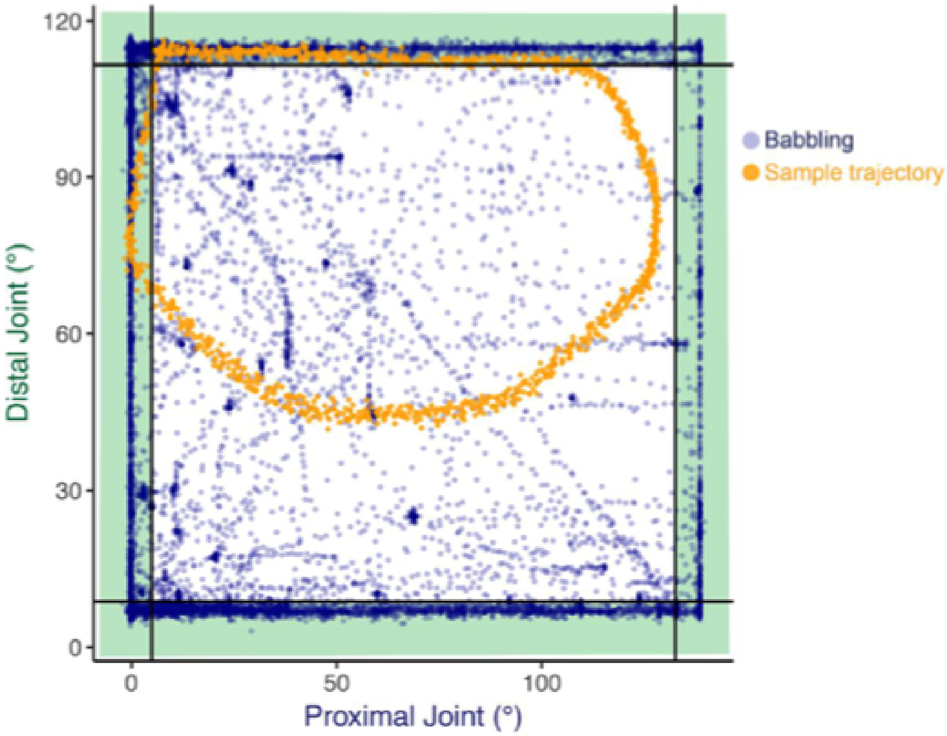


Figure 3.6: Distribution of joint angles visited during motor babbling vs. those used to produce a free cyclical movement in air - Motor babbling is done under no supervision, and in this tendon-driven double-pendulum primarily results in movements that rapidly fly towards the extremes of the ranges of motion of each joint (84.3% lie in the shaded within 5% of the joint limits (black lines).]). In contrast, the desired movement trajectories require exploitation of the relatively unexplored internal region of the joint angle space (orange points are 15 repeated cycles of a given cyclical movement).

could have expected that movements that caused more propulsion would be more energetically costly as they do more mechanical work against the treadmill, yet we also do not see such a consistent trend diagonally towards the top right. This is not to say that the high-level controller can add energy minimization as an element of the cost—although it may jeopardize the ability of the limb to apply mechanical work to the treadmill. Energy consumption may be necessary to regulate dynamic tendon shortening and lengthening (i.e., internal strain energy) to produce proper kinematics—a consequence of the simultaneously over- and under-determined nature of tendon-driven limbs [148].

3.4.4 Limitations, opportunities, and future directions

For organisms, as for machines, there exists a trade-off between improving performance via practice as each attempt carries the risk of injury, fatigue, and wear of tissues (e.g. blisters, inflammation of tendons, stress fractures)—in addition to energy expenditure and opportunity cost (i.e., spending time refining one task precludes learning a different one in a zero-sum lifespan). The G2P algorithm is designed to yield reasonable—if suboptimal— performance with limited data and no real-time feedback, but where the system continues to learn from each execution of the task. But it is also amenable to goal-driven refinements as each solution can serve as a starting point for subsequent optimization or improvements via feedback-driven corrections (PID, Recurrent Neural Networks, etc.).

Our fundamental motivation is to replicate how biological systems learn to move in a well-enough fashion when they must also limit the number of attempts using their own bodies. Our biologically-plausible system, in both its algorithmic and physical implementation, can also provide insight into tenable biological mechanisms that enable vertebrates to learn to use their bodies while mitigating the risks of injury and overuse—and yet successfully engage in natural selection and predator-prey interactions—which are the Darwinian arbiters in evolutionary success. The ingredients and steps of G2P are all biologically-tenable (i.e., trial-and-error, memory-based pattern recognition, Hebbian learning, experience-based adaptation [148]), and allow us to move away from the reasonable, yet arguably anthropocentric and teleological, concepts dominating computational neuroscience such as cost functions, optimality, gradients, dimensionality reduction, etc. [1, 148, 92, 57]. While those computational concepts emphasizing optimality are good metaphors, it has been difficult to pin down how one would be able to actually demonstrate their presence and implementation in biological systems45. In contrast, G2P can be credibly implementable in biological systems. Our own future direction is to demonstrate its implementation as a neuromorphic neuromechanical system, as we have done for other sensorimotor processes [95] as well as developing and modulating the features of more complicated behavior (such as locomotion) by adding some other hyperparameters to control features such as step-frequency, stride-size, etc.

3.5 Materials and Methods

In this section, we first introduce the control problem by describing the governing dynamics. Next, we go deeper into our learning and control algorithm (software). Finally, we finish this section by providing insight into the physical design of our physical system.

3.5.1 System dynamics

Equ. 3.1 defines the relationship between the joint kinematics and the applied torques of the limb [131] (forward model):

$$\ddot{q} = -I(q)^{-1}C(q,\dot{q}) + B\dot{q} + I(q)^{-1}T \tag{3.1}$$

where $q \in R^{2\times1}$, $\dot{q} \in R^{2\times1}$, and $\ddot{q} \in R^{2\times1}$ are joint angle vector and its first and second derivatives, respectively, $I \in R^{2\times2}$, $C(q,\dot{q}) \in R^{2\times1}$ is Coriolis and centripetal forces matrix, $B(\dot{q}) \in R^{2\times2}$ is the joint friction matrix, and $T \in R^{2\times1}$ is the applied joint torque vector. The musculotendon forces (here, cables pulled by the motors) are then related to the applied joint torques vector as described in Equ. 3.2:

$$T = M(q)F_0 a \tag{3.2}$$

where $M(q) \in R^{2\times3}$ is the moment arm matrix, F_0 is a 3×3 diagonal matrix with the maximal force values that can be exerted by each actuator and $\alpha \in R^{3\times1}$ is the normalized actuation value of each actuator [148, 92]. Please note this is an under-determined system (3 input force values generate two torques) where there is redundancy in the production of net joint torques at each instant. However, because the system is driven by tendons that can pull but not push (and not driven by torque motors coupled directly to the joints, as is common in robotics), joint rotations also depend on the ability of the controller to pay out and reel-in tendon as needed, else the movement can be disrupted or the system be non-controllable, respectively (this is why we use back-drivable brushless DC motors and maintain tension in the tendons at all times). As such, these tendon-driven systems present the challenge of being simultaneously under-and over-determined [148, 92]. The presence of constant tension in the tendons and friction in the joints (which can be heard in our video, see Supplementary Video 1) help stabilize the system but also add a deadband for control of subtle movements.

The goal of the inverse map is to find the actuation values vector (a) for any given set of desired kinematics $(q, \dot{q}, \ddot{q}$ without using any implicit model and only from the babbling and task specific data. The mapping done by the ANN used in the lower-level control of this study is described in Equ. 3.3.

$$a = ANN(q, \dot{q}, \ddot{q}) \tag{3.3}$$

Finally, the higher-level controller (in the RL task) is in charge of exploring the kinematic space and converging to desired kinematic trajectories that yield high reward. While these equations are effective for describing and controlling systems, we designed

G2P's lower level control with the premise that only the joint dynamics were observable (while not being used in real-time), and that the only controllable element is α. As a consequence, our system does not have any direct a priori conception of the model structure or the constants that drive the dynamics; lower level control must infer those relationships using training data from babbling and refine them after each attempt using only task specific input-output data (without being provided with a desired or error signal while refining the map after each attempt).

3.5.2 Learning and control algorithm

Learning and control in this first implementation of the G2P algorithm happens at two levels: (i) inverse mapping and refinements (the lower-level control) and (ii) the reward-based reinforcement learning algorithm (the higher-level control). The lower-level is responsible for creating an inverse map that converts kinematics into viable control sequences (motor commands). The higher-level control is responsible for reward-driven exploration (reinforcement learning) of the parametrized kinematics space which are further passed to the lower-level control and ultimately run through the system.

3.5.3 Inverse mapping and refinements

The lower-level control relies on two phases. As system is provided with no prior information on its dynamics, topology, or structure, it will first explore it dynamics in a general sense by running random control sequences to the motors, which we call motor babbling. After 5 minutes of motor babbling, the system creates the initial inverse map using the babbling data and then further refines this map using data collected from particular task-specific explorations, which we refer to as task-specific adaptation. This transition from motor babbling to adaptation to a particular task is the reason we refer to this algorithm as General to Particular or G2P.

3.5.3.1 Motor Babbling

During this phase, the system tries random control sequences and collects the resulting limb kinematics. A Multi-Layer Perceptron (MLP) Artificial Neural Network (ANN) is trained with this input-output set to generate an inverse map between the system inputs (here, motor activation levels) and desired system outputs (here, system kinematics: joint angles, angular velocities, and angular accelerations). Although sparse and not tailored for any subsequent task of interest, data from these random inputs and outputs suffice for the ANN to create an approximate general map based on the system's dynamics.

Random activation values for the babbling The motor activation values (control sequences) for motor babbling were generated using two pseudo-random number generators (uniformly distributed). The first random number generator defines the probability for the activation level to move from one command level to another. This value was set to 1/fs and therefore, the activation values for each actuator will change on an average rate of 1Hz. The second number defines the activation level of the next state with sampling from a range of 15% (to prevent tendons from going slack; see Tendons subsection) to 100% activation. The resulting command signals were stair-step transitions in activations to each motor. Three command signals were created (using different initial random seed) which ran three motors during the motor babbling. It is important to note that these stair-step random activities are designed to explore general dynamics of the system and are not tailored for any tasks performed during this study (see Fig. 3.4.2).

Structure of the Artificial Neural Network The ANN representing the inverse map from 6-dimesional limb kinematics to 3-dimensional motor control sequences (Equ. 3.3) has 3 layers (input, hidden, and output layers) with 6, 15, and 3 nodes, respectively. The transfer functions for all nodes were selected as the hyperbolic tangent sigmoid function (with a scaling for the output layer to keep it in the range of the outputs). The performance function was selected as MSE. Levenberg-Marquardt backpropagation technique was used to train the ANN and weights and biases were initialized according to the according to the Nguyen-Widrow initialization algorithm. Generating and training ANNs were performed using MATLAB's Neural Network ToolBox (MathWorks, Inc., Natick, MA; see MATLAB's Deep Learning Toolbox—formerly known as Neural Network toolbox—documentation for more details).

3.5.3.2 Task based refinements

Motor babbling yields sample observations distributed across a wide range of dynamics, but still represents a sparse sampling of the range of state-dependent dynamical responses of the double pendulum (Fig. 3.4.2). As a result, this initial inverse map (ANN_0, Fig. 3.3.1) can be further refined when provided with more task-specific data. The higher-level control will initiate the exploration phase using ANN0. However, with each exploration, the system is exposed to new, task-specific data, which is appended to the database and incorporated into the refined ANN_K map (Fig. 3.3.1). This refinement is achieved by using the current weights as the initial weight of the refined ANN and training it on the cumulative data after each attempt. A validation set is used to stop overfitting to the train data. The weights will not be updated for a run if the performance over the validation deteriorates for 6 consecutive attempts (default settings for the used toolbox). The data to be used to train the ANN was randomly divided into train, test, and validation sets with 70%, 15% and 15% ratios, respectively. It is important to note that refinements can update the map's validity only to a point; if major changes to the physical system are experienced (changing the tendon routings or the structure of the system) the network would likely need to re-train on new babbling data. This could be manually performed or a threshold for feedforward error could be set to activate re-babbling. However, we found that motor babbling done strictly while the limb was suspended in air nevertheless worked well when it was used to produce intermittent contact with the treadmill to produce locomotion on the treadmill and there was no need to re-babble in this study unless a motor, tendon cable, or link was replaced.

3.5.4 The reinforcement learning algorithm for the treadmill task

A two-phase reinforcement learning approach is used to systematically explore candidate system dynamics, using a 10-dimensional feature vector, ultimately converging to a feature vector that yields high reward. Similar to the ideas used in [49, 27] we have simplified the task by parametrizing it to avoid having the RL agent explore all possible states in the motor activation space (and its corresponding kinematics space). We have used a 10-dimensional feature vector to create cyclical trajectories. The goal of the policy search RL here is to converge to a parameter vector that yields high reward (treadmill movement). The use of a lower-level control to learn the inverse map enabled us to use a policy-based model-free RL whole parameters are reduced to only 10 (feature vector). The system will start from an exploration phase (uniformly random parameter search) and once the reward passed a certain threshold, policy will change to a Multivariate Gaussian distribution based stochastic search centered on the feature vector that yielded the highest reward so far (see below). Please note that the ANN in the lower-level control only creates an inverse dynamical model between the motor activation values and the joint kinematics (and has no information regarding the terminal reward: moving the treadmill). The RL agent perceives this inverse model simply as a part of the environment. Therefore, this method should not be confused with model-based RL algorithms where the agent utilizes a model to find actions whose predicted reward is maximal.

3.5.4.1 Creating cyclic trajectories using feature vectors

At each step of the reinforcement algorithm, the policy must produce a candidate set of kinematics. We defined ten equally-distributed spokes (each 36° apart; see Fig. 3.3.1c) on the angle-angle space. We can then set the lengths (distance from the center) of each spoke to define an arbitrary closed path that defines angle changes, which remains a smooth, closed trajectory. The positioning of the spokes and center are defined by the range of the babbling data. These ten lengths of the spokes are the 10-dimensional feature vector. Using interpolation of these 10 locations, we yield an angle-angle trajectory, and derivate those points (equally spaced in the time domain) to get the associated angular velocities and accelerations, which fully describe joint kinematics in time domain. Using the inverse map (lower-level control) these 6-dimensonal target limb kinematics $(q, \dot{q}, \ddot{q})$ will be mapped into the associated control sequences. The produced control sequences (motor activation values) are then replicated 20 times and fed to the motors to produce 20 back-to-back repetitions of the cyclical movement. Repeating the task 20 times allows us to smoothen the effect of unexpected physical dynamics of the task (e.g., system noise, unequal friction values over the treadmill band, nonlinearities of the system, etc.) which might lead to fluctuations in reward. The features were bounded in [0.1-1] range for the treadmill task and [0.2-0.8] during the free cyclical movements experiments to provide more focused task specific trajectories.

3.5.4.2 Exploration phase

Exploring random attempts across the 10-dimensional feature vector space (uniform at random in [0.1-1]; Equ. 3.1) eventually will produce solutions which yield a treadmill reward. Exploration continues until either the reward is higher than a predefined threshold or stopped when a maximal run number is surpassed (a failure).

3.5.4.3 Exploitation phase

Once the reward passes the threshold, the system will select a new feature vector in the vicinity of the feature vector from a 10-dimensional Gaussian distribution, with each dimension centered at the threshold-jumping solution. Much like a Markov process, with each successful attempt, the 10-dimensional distribution will be centered on the values of the feature vector which yielded the best reward thus far. The standard deviation of these Gaussian distributions is inversely related to the reward (the distribution will shrink as the system is getting more reward). The minimal standard deviation is bounded at 0.03. This mechanism helps in converging to the behavior with higher reward and explore their vicinity in feature space (forming high reward habits) within reasonable time span but without any guarantee on finding global optima. This is analogous to vertebrate learning behavior which can form efficient functional habits that may not be optimal. The governing equations on generating the next feature vector to be executed by the higher-level control are described in Equation 4:

$$\bar{F} = \begin{cases} u(f_m, f_M) & R_b < rewardthreshold \\ \\ max(min(\mathscr{N}(F_b, \Sigma(R_b)), f_M), f_m) & Otherwise \end{cases} \tag{3.4}$$

where u, and N are Uniform and Gaussian distributions, respectively, $\bar{F}$ is the feature vector of the next attempt, f_m and f_M are the min and max bounds for each feature in the feature vector, respectively (0.1 and 1 in this test), R is the reward, R_b is the highest reward so far, F_b is equal to the feature vector which yielded R_b and $\Sigma(R_b)$ is described as:

$$\Sigma(R_b) = \sigma(R_b)I_{10} \tag{3.5}$$

where I_{10} is a 10 by 10 identity matrix, R is the reward, and σ is defined as:

$$\sigma(R_b) = (b - R_b)/a \tag{3.6}$$

where a and b are scaling and bias constants, respectively. Here we empirically selected values of a and b to 600 and 9000, respectively (Table 3.1). Please note that these values only change the deviation of feature which will have an impact on the exploration-exploitation trade off; we observed that the performance of the system is not very sensitive to these values (i.e., the system will find an acceptable solution as long as reasonable values are set for them).

Table 3.1: Pseudo code for the RL algorithm

```
while $R < Reward_threshold$
    $F_{bar} = Uniform[0.15, 1]$
    $R = execute(F_{bar})$
end
$F_{best} = F_{bar}$
$R_{best} = R$
for $i = 1 : 15$
    $F_{bar} = Normal(F_{best}, sigma. * Identity(10))$
    $F_{bar} = max(min(F_bar, f_M), f_m$
    $R = execute(F_{bar}$
    if $R > R_{best}$
        $R_{best} = R$
        $F_{best} = F_{bar}$
        $sigma = (a - R_{best})/b$
    end
end
```

Between every attempt, the ANN's weights are refined with the accumulated dataset (from motor babbling and task-specific trajectories) regardless of the reward or reinforcement phase. This reflects the goal for our system to learn from every experience.

3.5.5 Simulations

We first prototyped our methods in simulation using a double pendulum model of a tendon-driven limb (equations and code for the double pendulum simulation are adapted, with modifications, from [122]). Similar to the physical system, our method proved to be efficient in the simulation and yielded comparable results (Figures S2 and S3). These simulations were kept isolated from the physical implementation, and its results were never used as seeds for the physical implementation. It is important to note that similar to any other modeling attempt, these simulations are simplified representations of the real physics. In addition, some values of the system are very challenging (if not impossible) to measure (e.g. the moment arm value function; which is another reason on why we think model-free approaches are an absolute need in this field). The simulations in this study are mainly designed to test the feasibility of the algorithm before testing it on the real system and are meant to only reflect the general structure of the system and parameters of these simulation are not fine-tuned to accurately mimic the physical system.

3.5.6 Physical system

We designed and built a planar robotic tendon-driven limb with two joints (proximal with a fixed height, and distal) driven by three tendons, each actuated by a DC brushless motor. A passive hinged foot allowed natural contact with the ground. We used DC brushless motors as they have low mechanical resistance and are backdrivable. The motor assembly and proximal joint are housed in a carriage that can be lowered or raised to a set elevation for the foot to either reach a treadmill or hang freely in the air (Figure 3).

We used the minimum number of tendons required to have full control on both joints (a minimum of n+1 tendons are required where n is the number of joints) [148]. Further considerations and part details can be found in the Supplementary Materials. Feasible Wrench Set and Design Validation The feasible force set of a tendon-driven is defined by all possible output force vectors it can create. Equation 3.7 describes the static output wrench for a tendon-driven system [148].

$$w = J(q)^{-T} M(q) F_0 a \tag{3.7}$$

where w represents the wrench (Forces and Torques) output and $J(q)^{-T}$ represents the Jacobian inverse transpose of the limb which transforms net joint torques into endpoint wrenches.

By evaluating all binary combinations for the elements in a, the resultant wrenches give rise to a feasible force set. It is important to preserve the physical capability of the tendon routing through the many iterations of limb design, so at each design phase we computed these sets for different positions throughout the limb propulsive stroke. Joint moment arms and tendon routings were simulated and ultimately built to have adequate endpoint torque and forces in all directions which is important for versatility [148]. Many other effective designs (different tendon routings, different link lengths, etc.) or design optimization techniques can be used and their performances in the tasks performed here can be evaluated; however, that is out of the scope of the current study.

3.5.6.1 Mechanical considerations

The carriage was attached to a wooden support structure, via linear-bearing and slide rails to adjust its vertical position. A clamp prevented sliding once the vertical position was set. Sandpaper was glued to the footpad and in strips across the treadmill to improve traction (Fig. 3.3.1 a and b).

3.5.6.2 Data acquisition

The control system had to provide research-grade accuracy and consistent sampling to enable an effective hardware test of G2P. A Raspberry Pi (Raspberry Pi Foundation, Cambridge, U.K.) served as a dedicated control loop operator—issuing commands to the motors, sensing angles at each of the proximal and distal joints, and recording the treadmill band displacement (Figure 3 a and b). Furthermore, the electrical power consumption for each motor was measured at 500Hz using current-sensing resistors in parallel with the motor drivers, calculating the watt-hours over each inter-sample-interval, and reporting the amortized mean power (watts) for the entire attempt. All commands were sent, and data received, via WiFi communication with the Raspberry Pi as csv files.

3.5.6.3 Running the system

The limb is placed in a consistent starting posture before activations are run to minimize variance in the initial conditions of the physical system. To aid development, a live-streaming video feed was designed for real-time visualization on any computer on the network (See Supplementary Video 1). A computer sends a control sequence to the Raspberry Pi, and after it is successfully run, the computer receives (i) the paired input-to-output data in csv format for iterative analysis or training, (ii) the net distance (mm) covered over the course of the entire action, and (iii) the amortized power the system consumed during the trial. Once data are collected, to calculate kinematics to train the inverse map, samples are first interpolated using their corresponding time labels to combat the nonuniform inter-sample interval of 78±5Hz. Prescribed activation trajectories are also served at this rate. The pipeline for data acquisition was designed with Python 3.6.

3.5.7 Data and Code Availability Statement

The source code can be accessed at https://github.com/marjanin/Marjaninejad-et.-al.-2019-NMI. Also, all other data (run data for experiments as well as the 3D printing files) can be accessed at:

https://drive.google.com/drive/folders/1FO0QJ2fBsdYCJs-h1LH7Iwb-wa0VPDi-?usp=sharing.

Chapter 4

Autonomous Control of a Tendon-driven Robotic Limb with Elastic Elements Reveals that Added Elasticity can Enhance Learning

Ali Marjaninejad[1], Jie Tan[2], and Francisco J. Valero-Cuevas[1,3]

[1]Department of Biomedical Engineering, University of Southern California, Los Angeles, CA

[2]Google Brain, Mountain View, CA

[3]Division of Biokinesiology and Physical Therapy, University of Southern California, Los Angeles, CA

4.1 Chapter summary

Passive elastic elements can contribute to stability, energetic efficiency, and impact absorption in both biological and robotic systems. They also add dynamical complexity

which makes them more challenging to model and control. The impact of this added complexity to autonomous learning has not been thoroughly explored. This is especially relevant to tendon-driven limbs whose cables and tendons are inevitably elastic. Here, we explored the efficacy of autonomous learning and control on a simulated bio-plausible tendon-driven leg across different tendon stiffness values. We demonstrate that increasing stiffness of the simulated muscles can require more iterations for the inverse map to converge but can then perform more accurately, especially in discrete tasks. Moreover, the system is robust to subsequent changes in muscle stiffnesses and can adapt on-the-go within 5 attempts. Lastly, we test the system for the functional task of locomotion and found similar effects of muscle stiffness to learning and performance. Given that a range of stiffness values led to improved learning and maximized performance, we conclude the robot bodies and autonomous controllers—at least for tendon-driven systems—can be co-developed to take advantage of elastic elements. Importantly, this opens also the door to development efforts that recapitulate the beneficial aspects of the co-evolution of brains and bodies in vertebrates.

4.1.0.1 Author Contribution

A. M. has designed experiments, wrote the code, run simulations and written the first draft of the manuscript. F.V-C. and J. T. have provided guidance and feedback to the manuscript and during each step of the experiment design and interpreting the results.

4.2 Introduction

Elastic elements are known to contribute in a passive way to a number of advantageous mechanical properties of robotic and biological systems. These include absorbing impacts, storing energy and postural stability. By absorbing impacts, elastic elements reduce noise and prevent damage to the structural elements and actuators (linkages, hinges and motors in robots; and bones, joints, and musculotendons in animals) or the environment [2, 154, 94, 55]. Also, opposing pairs of elastic elements act like proportional controllers (that can only pull) that can passively grant postural stability [124, 158, 5, 98, 102, 114, 121, 120]. It is also known that great energetic efficiency can be achieved by storing and timely release of energy in elastic elements [94, 138, 48].

These benefits, however, come at a cost. They can add nonlinearities, hysteresis and oscillatory modes to the system dynamics and, in general, make it harder to model and find accurate and robust analytical control solutions [144]. This is especially the case for analytical control methods that require precise models of the plant and the environment to operate accurately [108, 23] which is, in general, infeasible for most real-world plants and problems. Moreover, the mechanical properties of elastic materials are more often susceptible to changes in environmental (e.g., temperature), and use-case (e.g., wear and tear) factors.

An alternative approach to the control of plants with elastic elements would be to use control methods that do not depend on prior models, are data-driven, autonomous, or adaptable on the fly. However, the performance of these methods in dealing with added dynamical complexities introduced with the elastic elements has not been thoroughly explored. Moreover, the robustness of such methods to changes in stiffness values or operation in different functional regimes (e.g., nonlinear springs) needs to be addressed as well. This is an under-studied problem especially on bio-inspired, tendon-driven systems.

Tendon-driven systems are particularly interesting because they can offer great functional agility and versatility and freedom of design (e.g., actuator placement and tendon routing) [92, 86, 148, 62, 142]. Moreover, they can help us better understand and even approach the diversity and functional versatility of animals by shedding some light on governing principles of vertebrate form and function [90, 87].

These systems, on the other hand, are harder for engineers to model and analytically control for a number of reasons. To begin with, they are simultaneously under- and over-determined as, respectively, multiple muscle forces can produce a same net torque at a joint, yet a single joint rotation sets the lengths of all muscles that cross it. Thus, it can be challenging to find solutions that satisfy all the constraints imposed by tendons and by task specifications at the same time [90, 148]. Moreover, the fact that their actuators are not directly operating on the degrees of freedom (as is the case in joint-driven systems), makes it challenging to use an off the shelf controller (such as a simple PID setup) without having access to dynamical equations of the system or a forward or inverse kinematics model [91]. Also, these tendon-driven systems often require accurate modeling and control strategies for applications such as animation of life-like figures [74], control of anatomical limbs to understand neurological conditions [70, 110, 57], or functional electrical stimulation of limbs (e.g., [80] or [119]).

Here, we explored the efficacy of autonomous learning and control on a simulated bio-plausible tendon-driven leg across different tendon stiffness values. For the sake of generality, in this first study, we used two autonomous learning algorithms—one that builds a data-driven explicit kinematics model of the limb vs. one that uses end-to-end learning (see Methods)—to gauge the effect of elasticity of the actuators on learning and performance. Our results show that autonomous learning (both with an explicit inverse map and end-to-end) could learn to control the limb across all stiffness values. Our results also show that an appropriate value of added stiffness can enhance the learning and precision in all cases and even exhibit emergence of lower energy consumption. This is of great significance because the elasticity that is inherent to some types of plants (i.e., tendon-driven systems) can now be leveraged to improve learning and performance.

4.3 Methods

In this paper, we studied how adding elastic elements affects autonomous learning in a two-joint three-tendons simulated limb (similar to the physical system introduced in [90, 91]) in the MuJoCo environment [146](Fig. 4.1.a). The muscle model we use consist of a contractile element with Force-Length-Velocity properties [146, 148], a small parallel damper (100 Ns/m) and a parallel elastic element with stiffness value 'K' (see Fig. 4.1.b).

Specifically, we studied the convergence of the inverse kinematics map, how its performance accuracy changes with stiffness, as well as its adaptability when learning with one stiffness value and then performing using a different value.

As for learning, we used our autonomous few-shot hierarchical learning algorithm General-to-Particular (G2P) [90], and the end-to-end Proximal Policy Optimization (PPO) autonomous learning algorithm [137]. G2P is a hierarchical autonomous learning algorithm that, on its lower level, creates an inverse kinematics map using output kinematics collected from an initial random set of actuation commands (motor babbling). Systems that use an explicit kinematics model are, in general, easier to study and interpret, more data efficient and can generalize to a wider range of tasks; however, they can suffer from inaccuracies in the model especially during complex dynamical interactions (e.g., contact dynamics, injury to the body, or changes in the environment) [91, 90, 72, 20, 155, 107]. Systems that perform end-to-end learning (such as PPO), on the other hand, usually require larger number of samples to learn to perform a task, are harder to interpret due to their implicit modeling, and usually cannot generalize well across tasks [77, 136, 137, 45, 100]. These methods, however, can achieve better asymptotic performance even in challenging tasks.

4.3.1 Simulated experiments

For this study, we have performed three set of simulated experiments. In all simulations, elastic elements are considered as parallel elements with each musculotendon

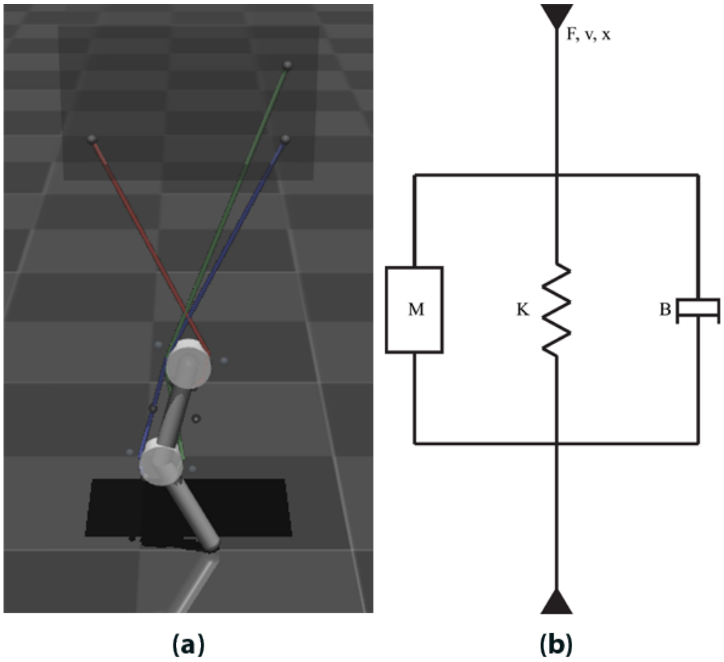


Figure 4.1: (a) The studied tendon-driven limb in MuJoCo environment. (b) each musculotendon consists of a muscle model (M), elastic element (K), and a damper (B).

(Fig. 4.1.b); the stiffness value of all elements are equal for each simulation and referred to as “stiffness”. The details for each of these set of simulations are provided below.

4.3.1.1 Controlling the limb with different stiffness values in the muscle model

In this simulation, for each stiffness value, we first randomly activated muscles and recorded the resulting kinematics (motor babbling [90]) for 3 minutes (100 samples per second). The recorded kinematics are joint angles, angular velocities, and angular accelerations for both joints (a vector of 6 values). Next, we trained a Multi-Layer Perceptron (MLP) Artificial Neural Network (ANN; one hidden layer with 15 neurons; trained for 20 epochs; 80% training 20% validation; loss function: MSE, optimizer: ADAM) with kinematics as input and activations as output to form the inverse kinematics map (similar to [90, 91]). Finally, this inverse map was used to control the system to perform two tasks: Cyclical and Point-to-point movements.

Cyclical movements In this task, the system was prescribed to move to generate a perfect circle in its configuration space (joint angle space). I.e., Joint angles change sinusoidal with $\pi/2$ phase difference. The frequency of these cyclical movements was set to 0.7 Hz and the task was continued for 21 cycles (total of 30 seconds).

Point-to-point movements Unlike the cyclical movements task, which is a smooth continuous task, the point-to-point task is consisted of discrete joint angle locations connected with rapid movements. In this task, 10 independent random angles (sampled from a uniform distribution within the range of each joint) are selected for each joint. The system then is commanded to go to each joint angle pair and stay there for 3 seconds (total of 30 seconds).

Similar to our previous work [90], we chose these tasks since they cover both extremities in the movement spectrum between continuous and smooth movements and discrete movements with fast transitions. For each joint, we calculate the error as the Root Mean Square Error (RMSE) of the difference between the joint angle and the desired angle in Radians. We disregard the error for the first 25% of the signal to make sure any initial condition effect is washed out [90, 91].

4.3.1.2 Adaptability to changes in stiffness

Stiffness value of an elastic element can change as a function of many physical factors such as temperature, wear and tear, etc. This can potentially endanger performance of the autonomous control of a system even if the system performs accurately in absence of any changes. This task is designed to study this effect as well as studying the feasibility of adaptive learning on-the-go (without a need to stop the system and redo the babbling) to compensate for these changes.

Here, we first perform the motor babbling for a system with an initial stiffness value (let's call it "A") and train the inverse map with the collected data. Then, we change the stiffness value (to let's say "B") and command the system to perform a cyclical movement attempt (described above). After each attempt, we concatenate all collected data and refine the inverse map using the cumulative data (refinement phase of G2P [90]). Here, we are showing results for up to 5 refinements for a system the stiffness value of which has changed (from "A" to "B") as well as systems that performed both babbling and refinements with the same stiffness value ("A" to "A" and "B" to "B") to provide better insight for a better comparison of the adaptation performance.

4.3.1.3 Functional task of locomotion

Studying the ability of our system in creating an inverse kinematics map for different stiffness values provides great insight into understanding the effects of stiffness on control and learning. However, a precise inverse map does not necessarily mean better performance in performing functional tasks that also features contact dynamics [91]. Also, most autonomous control methods do not use an explicit inverse kinematics map. Therefore, it is important to study the effects of stiffness on the performance of the system for a functional task. We chose a locomotion task that entertains contact dynamics, deals with gravity and inertia, and yields a reward as a measure of success. For this task, the limb is connected to a chassis that can move in x-axis (forward-backward) with friction to stop the system from floating. The system can also move on y-axis (up-down) where it is assisted with a spring-damper mechanism (similar to a gantry [91]). Please see the Supplementary Video (Supplementary Information section) for the task in action.

We have performed this task with two leading algorithms in autonomous learning. First, the G2P algorithm [90, 91], which is specifically designed to handle challenging task of learning and adaptation with no prior model and only using limited experience (which is a need in most real-world applications) and has proved to work well on the tendon-driven systems. Second, we have chosen the PPO algorithm, which is one of the leading end-to-end learning methods: for each observation, predicts activations that will yield high reward. The G2P implementation was in a faithful manner to the original paper [90], babbling time was selected to be 3 minutes, and the exploration-exploitation reward threshold was set to 3 meters of the chassis movement in the forward direction. For PPO, we used the "PPO1" implementation from Open AI's stable baselines repository. We run the training for 5000 episodes (1000 samples each; sampling rate: 100Hz)

4.4 Results

4.4.0.1 Controlling the limb for different stiffness values

Fig. 4.2 shows the MSE over the training data as a function of the epoch number across stiffness values. We see a consistent pattern in the training error curves in which systems with higher stiffness values start with larger error, yet once enough training rounds (epochs) are performed, they exhibit the smallest training errors. This pattern can be explained by the fact that more stiffness will add more dynamics to the system which initially makes it harder for the ANN to catch, but once converged, these extra dynamics can reduce the size of the solution space [17, 92, 148] (less ambiguity caused by the under-determined nature of the system) and therefore make more precise predictions. However, these MSE values only show how well the ANN could fit to the training data coming from the motor babbling (see Methods). Therefore, to study its performance across tasks, we now focus on the results collected from the cyclical and point-to-point tasks. Fig. 4.3 shows RMSE values for this simulation across all tested stiffness values (also see the Supplementary Video). We see that stiffness in the range of 2k-10k N/m can significantly improve performance compare to zero stiffness or very high stiffness

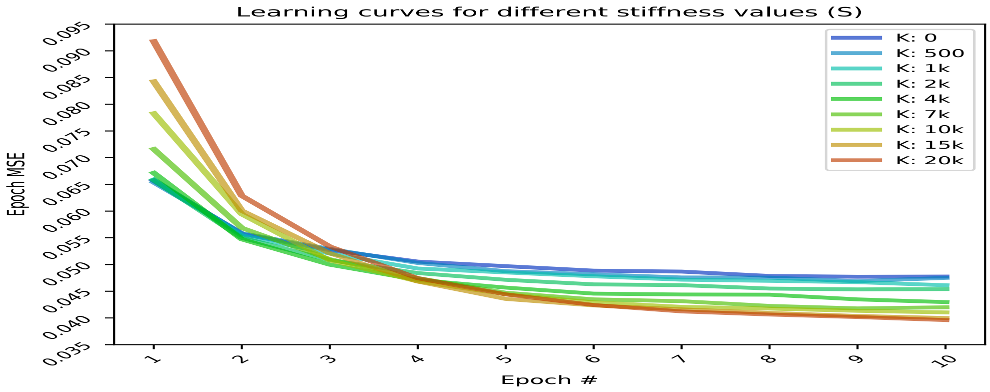


Figure 4.2: MSE over the training data as a function of the epoch number across stiffness values (Average of 50 Monte Carlo runs).

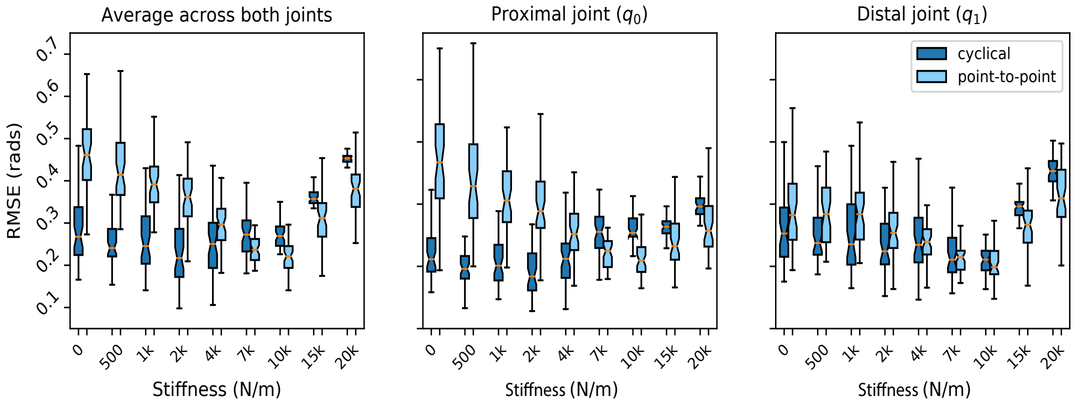


Figure 4.3: RMSE of joint angles as a function of stiffness for cyclical (dark blue) and point-to-point (light blue) tasks. 50 Monte Carlo runs for each case.

values. This improvement is even more significant for the point-to-point task which is explained by the fact that this task is more prove to the adverse effects of control in under-determined systems (see Discussion and [92, 148]).

4.4.0.2 Adaptability to changes in stiffness

Fig. 4.4 shows the performance of the system trained and tested with different stiffness values as well as its progress through refinements. Fig. 4.4 also shows the performance

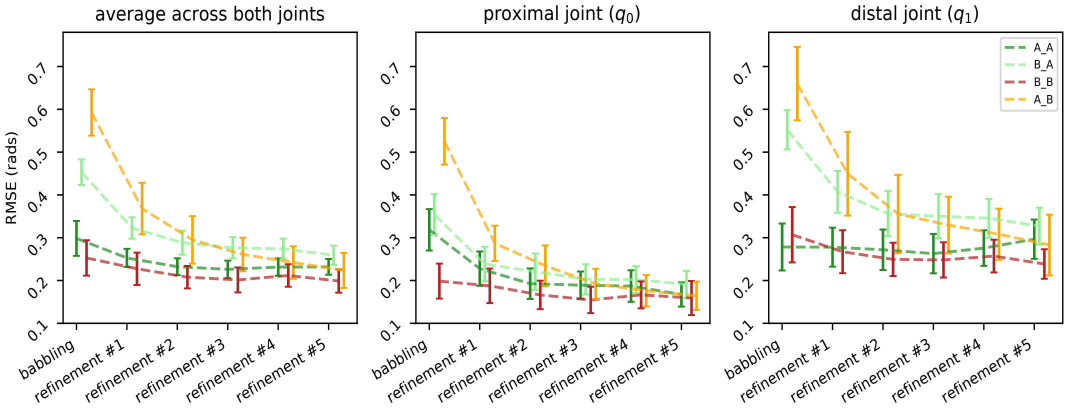


Figure 4.4: RMSE of the systems trained and tested with different stiffness values. A: 7K N/m and B: 2K N/m. (A_B (orange): trained with A, refined and tested with B; B_A (light green) the other way around), as well as the performance of systems trained, refined, and tested with the same stiffness values for baseline comparison (A_A and B_B, red and dark green, respectively), 50 Monte Carlo runs for each case.

of systems trained, refined, and tested with the same stiffness values for comparison. In Fig. 4.4, A and B correspond to 7K N/m and 2K N/m, respectively. Adaptability between other stiffness values, in general, also followed the same pattern (in all error bars and error shades in this paper, end to end height of whiskers/shades are equal to one standard deviation of the data). Fig. 4.4 results show that it is feasible for a system to create an initial inverse map and then adapt on-the-go while converging to similar performance measures as if it did not have a change. This is important since it will prove the use of elastic elements that are subject to change due to physical features (temperature, wear and tear, etc.) to be feasible in real-world robotic systems. We used G2P here and showed the feasibility of adaptation on-the-go to the changes in the tendon stiffness values. However, we want to underline that other adaptive learning methods can also be used (e.g., [72, 20, 9]).

4.4.0.3 Functional task of locomotion

In this section, we study the results for the functional task of locomotion for two autonomous learning algorithms: A hierarchical one that creates and utilizes a map of inverse dynamics and one with end-to-end (actions to observable states) learning; Namely, G2P and PPO (see methods). It is important to note that the focus of this section is to study the potential effects and contributions of the elastic element on autonomous learning of the tasks in an unbiased manner and not maximizing performance (e.g., using feedback to minimize the error [91], finding the optimal solution or the most efficient one) or modifying the algorithms to do so.

Fig. 4.5 shows the results for the G2P implementation of the locomotion task for 50 Monte Carlo runs (also see the Supplementary Video). Fig. 4.5a shows the success rate (if the algorithm found a solution that passes the 3m threshold within 100 exploration attempts). This figure shows that except for very high stiffness values, the algorithm could find a way to fulfil the task. Fig. 4.5b shows the ultimate reward for the successful attempts. Since G2P algorithm is not strict on maximizing the reward (finds a good-enough solution within few attempt), we cannot see any big distinction between these final rewards. Fig. 4.5c shows the energy consumption for the attempts with the ultimate reward (here we define energy as the sum of squared activation values for all three muscles and across time). This figure shows that the energy consumption for the mid-range stiffness values is lower. It is important to note that we did not put an energy cost term in the reward and therefore, this pattern is an emergent feature of the physics of the system. This

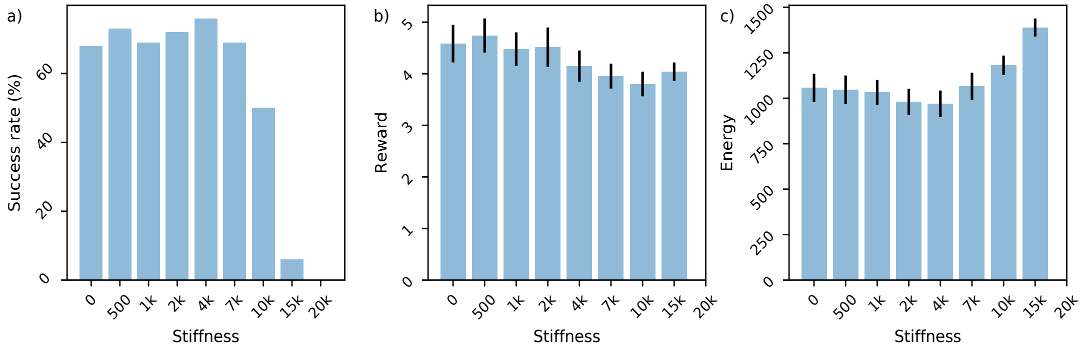


Figure 4.5: Results of the locomotion task using the G2P algorithm. 50 Monte Carlo runs for each case.

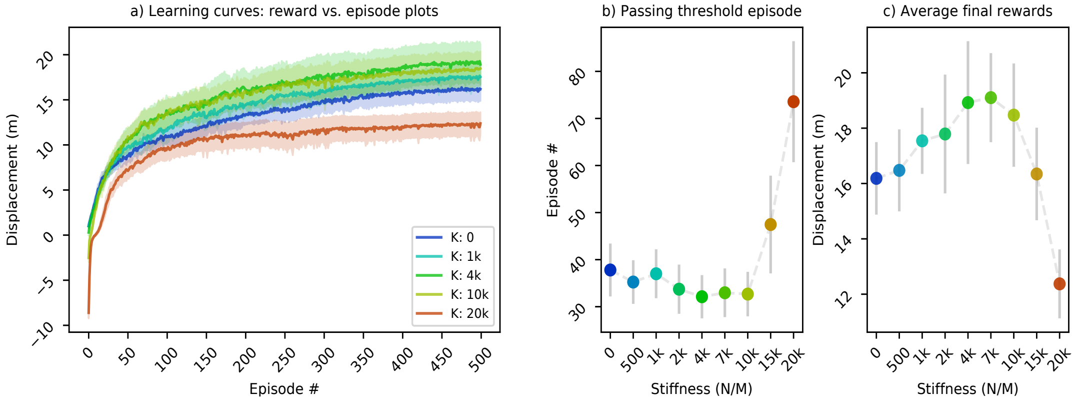


Figure 4.6: Results of the locomotion task using the PPO algorithm. 50 Monte Carlo runs for each case.

result justifies future studies that would focus on utilizing stiffness in reducing energy costs. Lastly, Fig. 4.6 shows results for the PPO implementation of the locomotion task for 50 Monte Carlo runs (also see the Supplementary Video; Negative values mean backward displacement). Fig. 4.6a shows that all learning curves exhibit a consistent pattern where systems with mid-range stiffness values raise faster and also end up with higher ultimate rewards. Fig. 4.6b shows the first episode in which the Fig. 4.6a curves passed an arbitrary reward cap (9m for this figure). The plot can slightly change based on the selected threshold, but the pattern is consistent in that systems with mid-range stiffness values need less episodes to pass any reward cap. Finally, Fig. 4.6c shows the ultimate rewards in which, again, consistent with all other findings of this paper, a mid-range stiffness value resulted in higher performance. It is important to note that although the PPO algorithm does not use an explicit inverse map, it builds an implicit inverse map which justifies why the results are consistent with the ones coming from the G2P algorithm, which uses an inverse map in a hierarchical structure.

One important point we observed in our simulations was oscillatory behavior (chatter) in systems with very high stiffness (see Supplementary Video). The likely origin of this is that high stiffness in the muscle model, we now see, can make the system have modes at higher resonant frequencies (analogous to high gains for small errors in a proportional controller) that can lead to instability and interfere with the numerical integrator. This happens at high stiffness values even though our MuJoCo model has mild damping and frictional losses distributed throughout the body (i.e., at joints, contact model, muscles, etc.) to make the system more stable, realistic and numerically efficient.

4.5 Discussion

Here we show, first, the feasibility of autonomous learning and adaptation in the presence of elastic elements in tendon-driven systems. And second, we provide evidence that changes in the parallel stiffness of the actuators (i.e., muscle model) affects both learning speed and performance. Our results are useful in that they show (i) fast learning and adaptation in systems known to be challenging to control with analytical approaches, and (ii) great promise and opportunity for the design of robotic systems where tuning the stiffness of the actuators can greatly enhance performance while leveraging the inherent passive properties of elastic elements that also grant stability and potential energy efficiency with, importantly, minimal to no degradation in learning speed.

These findings are critical for the future evolution of robot design, which to date has splintered into two main camps: ‘conventional’ robot design with rigid bodies and actuators [156] vs. ‘soft robots’ that have few to no stiff elements [152]. Our work here now points to a third option that can, in principle, combine the benefits of both approaches by populating the spectrum between them. In our prior work, we have emphasized that the design space of tendon-driven systems must include both the topology of the limb (i.e., the number, type and connectivity among its elements) and the parameters of the individual elements (e.g., joints, linkages and tendons) for both robots and musculoskeletal systems [52, 149]; we have also explored the extreme case of purely data-driven locomotion of tensegrity structures [130] and limbs [90]. However, that work did not explicitly explore the consequences of elasticity to learning per se.

We now argue that elasticity is an inevitable element of tendon-driven robots and biological systems (see Introduction), and thus must be systematically and explicitly considered in this current AI wave seeking to develop autonomous learning for robots, and to understand neuromuscular control in animals. As such, our results argue for, and enable, the co-development of robot bodies and autonomous controllers that take advantage of elastic elements, which can lead to improved learning and performance—while also taking advantage of its intrinsic benefits of stability, energetic efficiency, and impact absorption. It is important to underline that the main focus of this study was not to optimize for performance or energy efficiency. Moreover, we used two of the most recent algorithms that prove to be suitable for the test case in hand but similarly, other state of the art algorithms can also be used in the future to control these systems. Our results, therefore, open the door to development efforts that recapitulate the beneficial aspects of the co-evolution of brains and bodies in vertebrates.

One particularly interesting observation from Fig. 4.2 is that it was initially easier for the ANN in G2P to fit to the data when the muscles had low stiffness values. And then, after a few epochs, the fit was better with higher stiffness values. This suggests that, in principle, learning would be optimized if one were to start out with low stiffnesses that increased over time. This is paralleled by the fact that most vertebrates start their life with a more compliant anatomy which stiffens with development [141, 115, 37]. In our prior work, we have discussed in detail how the over-determined nature of tendon-driven systems with stretch reflexes in the muscles can make them difficult to control [148]. This is because the rotation of a joint will be impeded or disrupted if even one of the muscles that crosses it fails to lengthen (via its stretch reflex). That is, multiple constraints (i.e., lengthening of muscles) must be satisfied when driven by few variables (i.e., join angles). Such over-determined systems, which have more variables than equations, have at most one solution and are solved in practice via least-squares error methods. In such methods, a solution is found by finding a set of variables that violate the constraint equations the least (in a Euclidean norm or sum-of-squares sense). This is why, in the past, we have called the elasticity of musculotendons (the combinations of muscle and tendon) as a 'critical enabler' of the neural control of smooth movements [148]. The results Fig. 4.2 bear this out: it is easier to learn to control tendon-driven system where low stiffnesses at the muscles provide a large error margin for muscle lengths at the expense of performance; but stiffening the system once the initial learning has taken place will improve performance. This, in a sense, is a form of morphological curriculum learning that can enable new thinking about 'developmental robotics,' where changes that happen within an individual's life span improve learning and performance, echoing the work of Bongard where morphological changes within a single individual aid learning [8]. This is an interesting path for future work and is distinct from 'evolutionary' robotics that occurs over multiple generations of individuals.

Other future work could focus on the development of hardware/software to exploit these benefits of elastic elements, especially in tendon-driven systems. This also opens up opportunities for testing autonomous learning algorithms and assessing their performance in more sophisticated designs (such as bipeds or quadrupeds, especially in their physical implementations), and more challenging tasks and environments. It is important to note that this study worked within the abilities and limitations of MuJoCo, which implements a very particular version of a Hill-Type muscle model that does not include a tendon with the elasticity and viscosity parameters of the aponeurosis and tendon [148].

The stiffness values that we changed in the muscle model are those for the parallel elastic element to the force generating module that uses a simple approximation to the force-length and force-velocity properties of muscle [146, 148], and does not contain the natural spinal closed-loop control (i.e., afferentation) of muscles [104]. Studying the effects of the series elastic element (which, by the way, is also the stress-strain curve of a mechanical cable in a robot) would be an interesting and necessary path to follow in the future work.

SUPPLEMENTARY INFORMATION

The code and the supplementary files (Videos, MuJoCo models, etc.) can be accessed through project's Github repository at:

https://github.com/marjanin/tendon_stiffness

ACKNOWLEDGMENTS

This project was supported by NIH Grants R01-052345 and R01-050520, award MR150091 by DoD, and award W911NF1820264 by DARPA-L2M program. Also, by USC Provost Fellowship to A.M.

Chapter 5

Simple Kinematic Feedback Enhances Autonomous Learning in Bio-Inspired Tendon-Driven Systems

Ali Marjaninejad[1], Darío Urbina-Meléndez[1], and Francisco J. Valero-Cuevas[1,2]

[1]Department of Biomedical Engineering, University of Southern California, Los Angeles, CA

[2]Division of Biokinesiology and Physical Therapy, University of Southern California, Los Angeles, CA

5.1 Chapter summary

Error feedback is known to improve performance by correcting control signals in response to perturbations. Here we show how adding simple error feedback can also accelerate and robustify autonomous learning in a tendon-driven robot. We have implemented two versions of the General-to-Particular (G2P) autonomous learning algorithm using a

tendon-driven leg with two joints and three tendons: one with and one without real-time kinematic feedback. We have performed a rigorous study on the performance of each system, for both simulation and physical implementation cases, over a wide range of tasks. As expected, feedback improved performance in simulation and hardware. However, we see these improvements even in the presence of sensory delays of up to 100 ms and when experiencing substantial contact collisions. Importantly, feedback accelerates learning and enhances G2P's continual refinement of the initial inverse map by providing the system with more relevant data to train on. This allows the system to perform well even after only 60 seconds of initial motor babbling.

5.1.0.1 Author Contribution

A. M. has designed experiments, wrote the code, run simulations and hardware experiments, written the first draft of the manuscript. D. U-M. has designed the physical system and contributed in hardware experiments and to the manuscript. F.V-C. has provided guidance and feedback to the manuscript and during each step of the experiment design and interpreting the results.

5.2 Introduction

The field of robotics in general would benefit greatly from autonomous learning to control movements with minimal prior knowledge and limited experience [90, 72, 103].

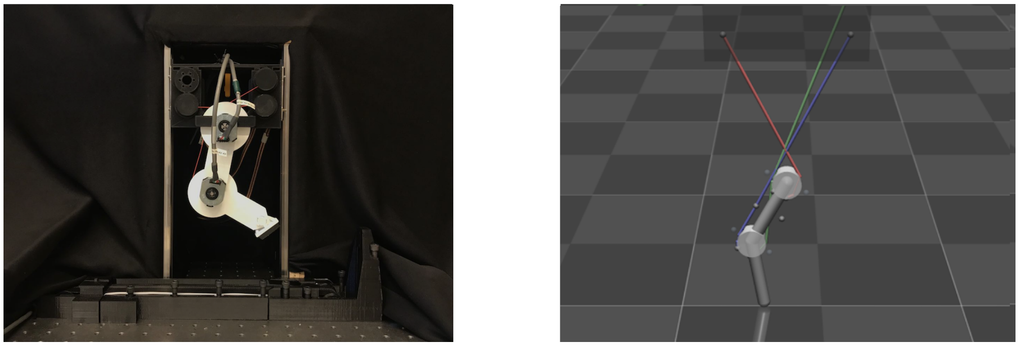


Figure 5.1: Physical tendon-driven robotic limb (left) and the simulated system in MuJoCo environment (Right)

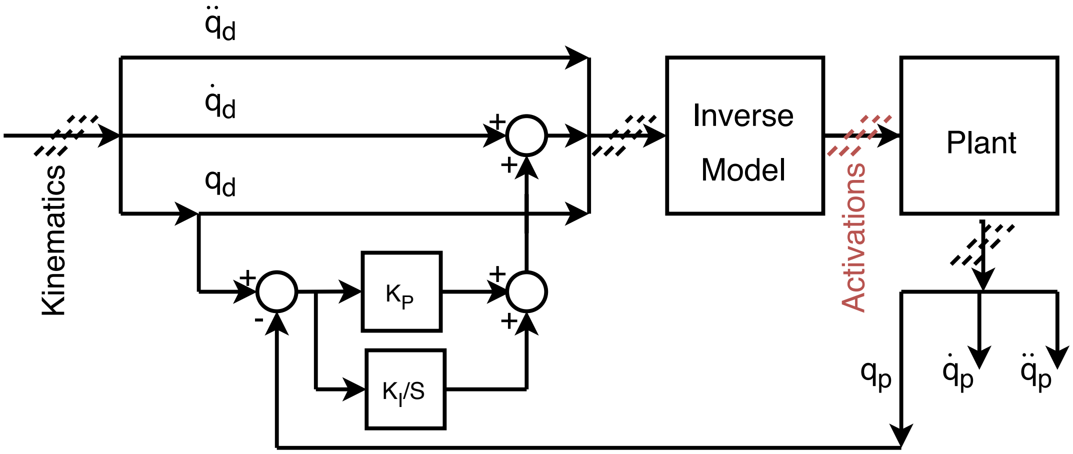


Figure 5.2: Schematic of the closed-loop system

Extensive trial-and-error experience in the real world can be very costly both in biological and robotic systems. Not only does it risk injury, but the opportunity cost can be large. Therefore, evolutionary game theory in biological systems favors systems that can function suboptimally or well-enough with only limited experience and continue to learn on-the-go from every experience [90]. Biological systems can then use sensory feedback to refine performance as needed.

Such learning from limited experience is also attractive in robotics [72, 90], mostly in situations where optimality is not as critical as adaptability to unstructured environments, unpredictable payloads, or working with systems for which creating accurate models is costly or time consuming. Thus, data-efficient learning that produces suboptimal behavior can be a practical and attractive control strategy as it does not rely on accurate prior models or extensive expert knowledge [65, 93, 101, 36, 33, 131, 47, 69] , or require thousands of hours of or learning in simulation [100, 143, 45, 3, 137, 136] (please see [90] for detailed discussion on how our General-to-Particular (G2P) algorithm relates to the field).

A drawback of learning with limited experience that produces suboptimal behavior is that the performance of the model can degrade when encountering dynamics far from those under which it was trained. On the other hand, systems that heavily depend on the feedback error correction would not perform efficiently and are prone to instability, especially in the presence of sensory delays[9, 90]. Moreover, it is important to note that in a tendon-driven system, actuation is not directly connected to the joint and therefore, a simple off-the-shelf PID controller cannot be used without knowing the dynamical equations of the system [90, 148, 92]. Thus, here we explore the combination of data-efficient learning algorithm, G2P, with simple feedback to maintain the key benefits of learning under limited experience while improving performance and robustness to perturbations (or unmodeled dynamics) as needed. This approach is directly inspired by biological systems that, under certain circumstances, successfully use simple corrective responses triggered by delayed and non-collocated sensory signals [99, 14, 13].

As an initial proof-of-principle, we implemented two versions of the data-efficient autonomous learning algorithm, G2P [90] (that is originally designed to control tendon-driven systems) : One purely feedforward (open-loop) as published in [90], and one with simple feedback on joint angles (closed-loop). Both implementations of G2P find motor commands that produce desired leg kinematics by creating an inverse map. The initial inverse map is generated from "motor babbling" input-output data (i.e., random sequences of input commands to the three motors driving the tendons that produce time histories of two joint angles of the leg). We find that the performance is, as expected, better for the closed-loop system as it compensates for errors in the leg joint angles arising from imperfections of the inverse map or external perturbations (e.g., contact dynamics). However, we also find that, by collecting more task relevant data, this simple feedback accelerates learning and improves the quality of the inverse map that enables the system to work with shorter motor babbling sessions and also improves learning on-the-go capability (the refinement of the inverse map from each experience) of the system.

In addition, we also report improved performance even when sensory signals are delayed. We have validated our method on a physical tendon-driven leg to demonstrate its utility in real-world applications.

5.3 Methods

In this section, we first discuss the design of the tendon-driven system. Next, we formulate and describe the controller design. Lastly, we discuss the tests we performed in detail.

5.3.1 Tendon-driven leg design

Tendon-driven anatomies are a relevant use case because they are difficult to control as they are simultaneously nonlinear, under-determined and over-determined [92, 148, 90, 17]. The simulated leg is a similar design to the physical leg used in [90]. It is a 2-DoF planar leg actuated with 3 (minimal number of tendons to control a 2-DoF leg) tendons. Unlike [90], the simulation model uses a Hill-type model of skeletal muscle (MuJoCo's built in force-length and force-velocity model [146, 148] and has moment arms that can bowstring. The physical system used for validation is a replica of the one used in [90] with an improvement on the data acquisition system (we use PXI system from National Instruments, Austin, TX, USA). Fig. 5.1 shows the tendon-driven legs for the physical and the simulation systems.

5.3.2 Controller design

Our system takes desired movement kinematics (joint angles for each joint and their first two derivatives; angular velocities and angular accelerations) and outputs activation values that will drive the actuators (skeletal muscles in simulation and electric DC motors connected to the tendons on the physical system) to produce the desired kinematics on the leg.

The feedforward path consists of an inverse mapping that maps the desired kinematics to the activations that will ideally (in the case of a flawless inverse map and without any perturbation) create activations required to replicate the desired kinematics on the plant (Fig. 5.2). This inverse map is created by training a Multi-Layer Perceptron (MLP) Artificial Neural Network (ANN) with 15 hidden layer neurons using the data collected during a short phase (5 minutes) of random movements and observing their corresponding kinematics which is called motor babbling (please see [90] or the supplementary code for more details on the feedforward path). Once the activations are calculated, they will be fed into the plant and the corresponding kinematics will be recorded. These observations can then be used in a feedback loop to compensate for any error in the inverse map or the error caused by external perturbation.

Here we only use joint angles as the real-time sensory feedback. Also, we mainly focus on reducing the error on the joint angles (as opposed to its derivatives) since error in joint angles is less forgiving than joint velocity or acceleration in successful compilation of most day to day tasks either being manipulation, locomotion or other movements;

however, if desired, user can substitute the error term with the angular velocity or acceleration or a weighted combination of them. We know that for a given joint, joint angle and angular velocity are related by equation 5.1.

$$\Delta q = \dot{q}\ .\ dt \tag{5.1}$$

where t is time. Therefore, we can compensate the error in position by changing the velocity corresponding to the magnitude and the direction of the error. We implement a PI controller like method where we add an adjustment term to the desired angular velocity of each joint proportional to its current and cumulative error (see discussions for alternative choices). Equations 5.2- 5.6 describe relationships between all system variables over a complete loop:

$$a[n]_{(3\times1)} = ANN(q_c[n]_{\,(2\times1)}, \dot{q}_c[n]_{\,(2\times1)}, \ddot{q}_c[n]_{\,(2\times1)}) \tag{5.2}$$

where $a[n]$ is the activation vector at time sample n and q_c, $\dot{q}_c$, and $\ddot{q}_c$, are joint angle, control angular velocity and control angular acceleration, respectively. These control variables are calculated as follows:

$$q_e = q_d - q_p \tag{5.3}$$

$$\dot{q}_a = K_{P\,(2\times2)}\ q_e + K_{I\,(2\times2)} \int q_e\ .\ dt \tag{5.4}$$

$$\dot{q}_c = \dot{q}_d + \dot{q}_a \tag{5.5}$$

$$\ddot{q}_c = \ddot{q}_d, \quad q_c = q_d \tag{5.6}$$

where subscripts d, p, e, and a stand for desired, plant, error, and adjustment, respectively. Also, K_P and K_I are diagonal matrices defining the proportional and integral coefficients for each joint. The complete schematic block diagram of the closed-loop system is depicted in Fig. 5.2.

5.3.3 Studied tasks

To demonstrate the performance of the proposed method and its capabilities, we have tested it in a number of different cases, each of which demonstrates at least one of its prominent features. Wherever applicable, we have compared the results with the ones produced by the open-loop method used in [90].

5.3.3.1 Cyclical movements in-air task

During this task, the leg is suspended in-air (i.e. no contact dynamics/external perturbations involved) and is commanded to perform 50 random cyclical patterns (10 cycle with 2.5 seconds each). Since there are no external perturbations applied to the system in this task, an ideal inverse map should be able to perform it flawlessly. However, the inverse map trained with limited experience is almost always imperfect; during this task we will study the effect of the proposed closed-loop system on reducing these imperfections.

These patterns are created by projecting a vector of 10 random values sampled from a uniform distribution ($U(0,1)$) into joint angle space as described in [90]. In short, each random number defines the normalized radial distance from the center of the joint angle space of one of equally distributed spokes (each 36 degrees apart) and then, these points will be connected and the resulting closed cycle will be filtered to make it smooth (see [90] for more details).

5.3.3.2 Point-to-point movements in-air task

Unlike the continuous and smooth cyclical task, point-to-point task is consisted of discrete ramp-and-hold movements. Since the inverse map was trained with random activations (motor babbling), it would be interesting to study how it performs when the desired task involves maintaining joint angles in specific positions (both angular velocity and accelerations will be equal to zero in all these positions). The point-to-point task is designed to study these cases and involved 50 trials where in each trial the leg is going to be commanded to go to 10 random positions ($U(joint_{min}, joint_{max})$ for each joint) and stay there for a predefined duration (2.5 seconds here).

5.3.3.3 Different cycle period durations task

During the motor babbling, the inverse map is introduced to a very sparse set of samples in the 6D kinematics space [90]. Although it has fully swiped across joint angles for both joints during the motor babbling phase, there are many combinations of these angles with their angular velocities and accelerations that will not be experienced. Here we are going to study the performance of the system for perfectly cyclical movements (sin and cos) over a wide range of cycle periods (1 / cycle frequency) to investigate how well the open-loop system performs in each case and to compare the performance of the proposed feedback controller.

5.3.3.4 Performance in the presence of contact dynamics tasks

Dealing with contact dynamics is a current challenge in robotics [25, 26]. Therefore, it is important to test the performance of the proposed method in the presence of contact dynamics. We have shown that the open-loop system can perform well when introduced to minor contact dynamics[90]; however, the performance of the system has not been studied under the effect of significant contact dynamics caused by the need to push the system forward/backward in the presence of an antagonist force or the need to carry its own weight (note that the system was trained in-air and therefore adding weight will be a major change to its dynamics). Here, we have studied the performance of the system during two tasks both including contact dynamics.

Locomotion with the gantry In this task, we have lowered the chassis (so that the leg can touch the floor) and let it move on the x-axis (forward/backward) with friction. Moreover, we have held it up with a spring-damper (build-in features in MuJoCo) so that it can partially compensate for the weight (similar to a gantry). Similar to the “Cyclical movements in-air”, here we have applied 50 different cyclical movements and studied the performance of the system. Please note that here we are simply applying random cyclical movements to compare open-loop and closed-loop performance; however, a higher-level controller can also be used to find better movement trajectories to yield higher forward displacement [90].

Holding a posture under a weight In this task, we took off the spring-damper mechanism provided by the gantry and increased the weight of the chassis significantly. The goal here for the leg is to stay vertically straight (standing leg position) while reacting to a strong downward force applied to it, due to the added weight of the gantry.

5.3.3.5 Learning from each experience task

Experience can be very costly in real-world physical systems [90] and therefore, an efficient system should be able to start performing as soon as possible and improving the performance with the data coming from each experience. During this task, we start with an inverse map created using a shorter duration (1 minute) and run the system on a cyclical task for 25 repetitions; after each repetition, we refine the inverse map with the cumulative data from all the experience that the system had so far (including the motor babbling). We repeat this process for 50 different cyclical trajectories.

5.3.3.6 Variable feedback delay task

Delay in the sensory feedback or processing information is inevitable in real-world applications. In a system that solely depends on error correction, these delays can inject large errors and even drive the system to instability. We have studied the performance of the system over a wide range of loop delays (from 5 to 100 ms; which is about the largest delay in the human sensory-motor loop) over 50 random cyclical movements.

All tasks were performed on both simulation (sim) and physical (phys) systems except ”performance in the presence of the contact dynamics” and ”variable feedback delay” tasks which were only performed on simulation due to physical limitations. Also, physical results for the ”learning from each experience task” has already been studied on [90].

5.4 Results

In terms of the error (root mean square error between joint angles to the desired reference trajectories; will be referred to as “error” from here on), as expected, we see the closed-loop control architecture reduces the error compared to the open-loop one in all cases. Fig. 5.3 shows the average error for the open-loop and the closed-loop system across all tasks.

5.4.1 Cyclical movements in-air task

Fig. 5.4 shows a sample trial of the cyclical movement in-air task for the physical system (also see supplementary information section for the supplementary video). This figure shows that the error is larger at the distal joint compared to the proximal joint. This is because all three tendons cross the proximal joint first, thus errors propagate to, and accumulate at, the distal joint.

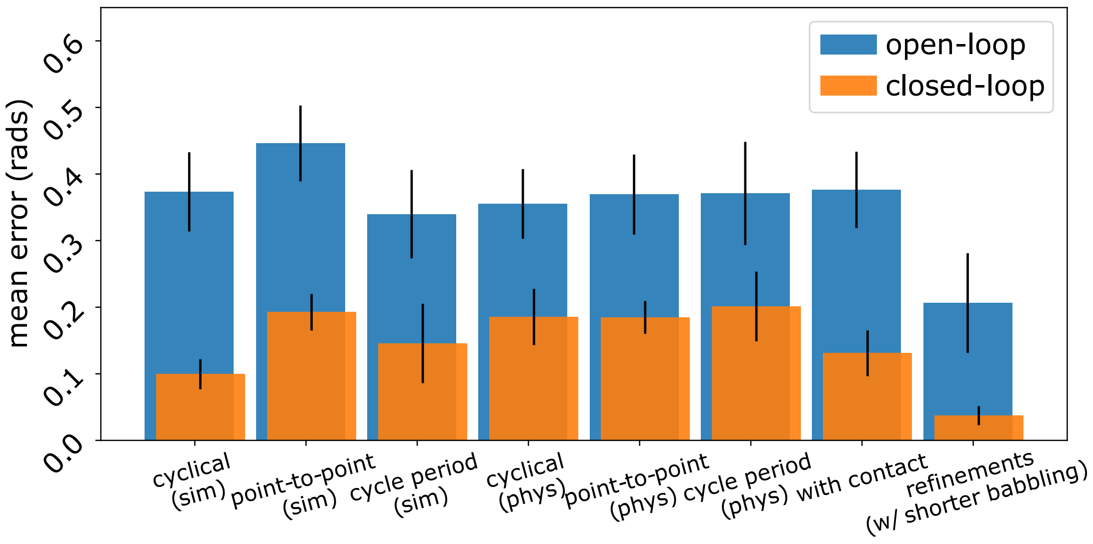


Figure 5.3: The average error for open-loop and closed-loop systems across all tasks.

For all "sample run" plots that were performed in both simulation and the physical system (Figs. 5.4, 5.5, and 5.7) we observe very similar patterns and are only reporting the physical system results here. The reader, however, can access all plots in [91].

5.4.2 Point-to-point movements in-air task

Fig. 5.5 shows a good example of the limitations of an open-loop system (also see supplementary video). When the system is commanded to go to a new position, it can do so—except in cases where the commanded change is small. The inverse map may not have sufficient resolution to implement such small changes. Also, please note that both angular velocities and angular acceleration inputs will be zero (except during the transitions which are very short) and the system needs to go to the right position based only on joint angle values. However, the closed-loop system detects and corrects those errors.

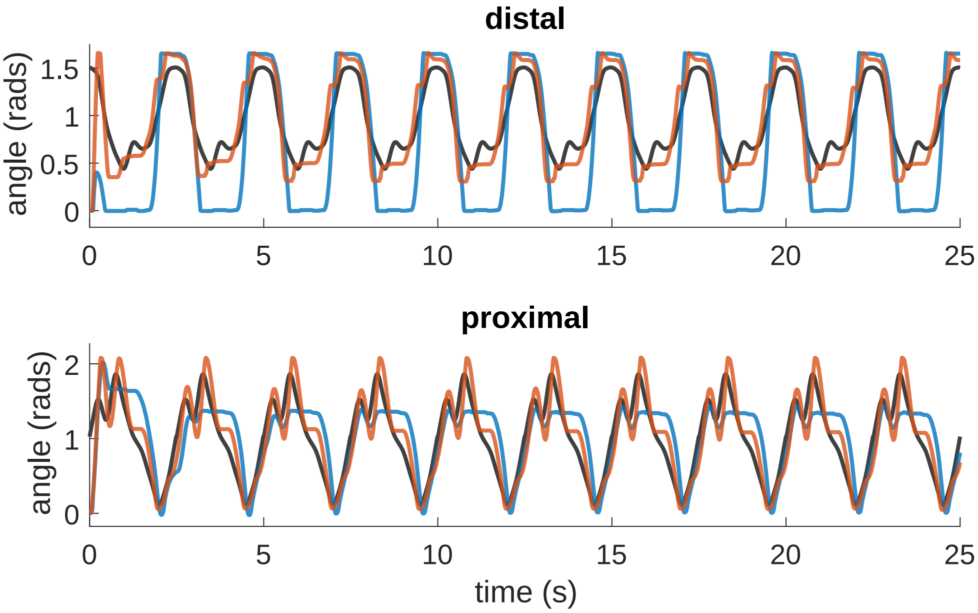


Figure 5.4: The desired (black), open-loop (blue), and closed-loop (orange) joint angles for one trial of the cyclical movements in-air task.

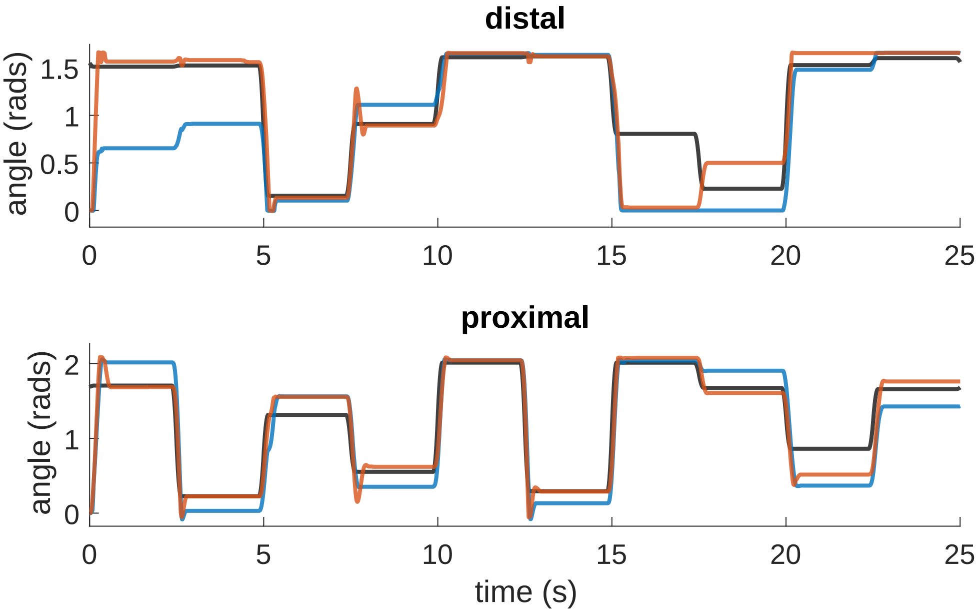


Figure 5.5: The desired (black), open-loop (blue), and the closed-loop (orange) joint angles for one trial of the point-to-point movements in-air task (over one sample run).

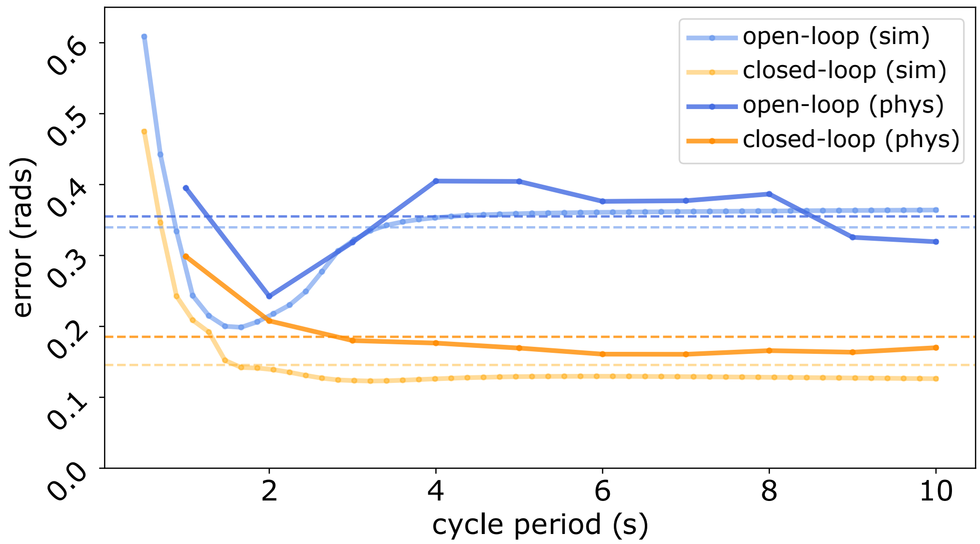


Figure 5.6: Error values for different cycle period durations task as a function of cycle duration for the open-loop and the closed-loop systems (sim: simulation; phys: physical system).

Importantly, this also improves the on-the-go training of the inverse map (see learning from each experience task). Note the unavoidable small fluctuations around the desired location, which are naturally caused by having a simple error correction feedback strategy. Better tracking can be achieved with more sophisticated closed-loop controllers, but that is beyond the objective and scope of this work.

5.4.3 Different cycle period durations task

The simple proportional-plus-integral feedback on joint angles has a limited bandwidth. We expect, and see, that its ability to correct errors degrades for cyclical movements with shorter cycle periods (i.e., higher frequencies). Fig. 5.6 shows improved performance of the closed-loop system for cycle period durations longer than ~ 2 seconds for both simulated and physical systems. As expected, the open-loop system also has problems at short cycle periods (due to effects of inertia, excitation of the nonlinearities of the double pendulum, bandwidth of the motors, etc), but the error does not improve as the cycle periods lengthen. The closed-loop system plateaus at a small average error of 0.1-0.2 radians per cycle quickly and then continues to reduce slowly. The error in the open-loop system, in contrast, has an error that is roughly twice as large with minimum at periods of $\sim 1.5-2$ seconds, which is perhaps closer to the region it experienced during babbling and also close to the system's resonant frequency. Fig. 5.7 shows the desired and actual outcomes of the task (for both open-loop and closed-loop systems) over one sample run for a 2.5 seconds cycle period (also see supplementary video).

5.4.4 Performance in the presence of contact dynamics tasks

5.4.4.1 Locomotion with the gantry

This leg system was designed to ultimately produce locomotion. Therefore, we tested in simulation how the closed- and open-loop systems performed this task. When introduced to mild ground contact (barely swiping the ground) both methods performed similarly well and comparably to the in-air task—albeit with a slightly larger error. However, when the simulated gantry was brought lower (and therefore more substantial contact dynamics were introduced), the open-loop system failed to clear the ground and could not complete the movement cycle to match the desired trajectories (see supplementary video and Figs. 5.3 and 5.8). In contrast, the closed-loop system was able to complete

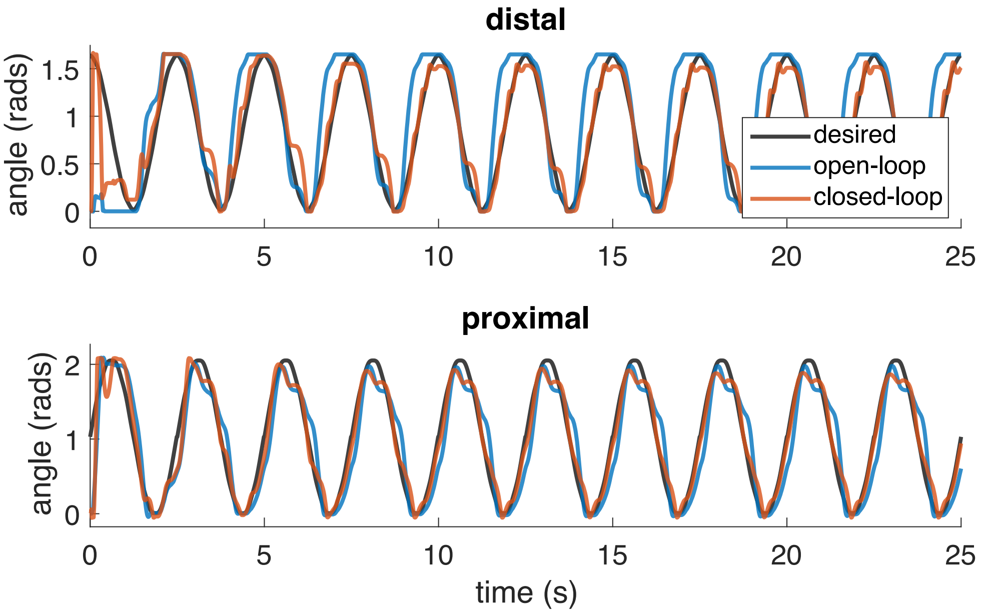


Figure 5.7: The desired (black), open-loop (blue), and the closed-loop (orange) joint angles for one sample run of the different cycle period durations task with a cycle period of 2.5 seconds.

the swing-phase and recover from the ground contact (see Supplementary video), which then resulted in very small errors even in the presence of these contact dynamics. This is expected as contact dynamics can be thought of as physical perturbations that were not included in the motor babbling. Thus, the open-loop system naturally performs poorly (even with a well-refined, accurate inverse map). However, it was important to see that even simple feedback was able to compensate for such strong unmodeled, perturbations.

5.4.4.2 Holding a posture under a weight

In our simulations, we also observed that the open-loop system cannot compensate when a weight is applied to the chassis of the leg (the leg collapses under the weight). In contrast, the closed-loop system compensates—as much as the strengths of the muscles

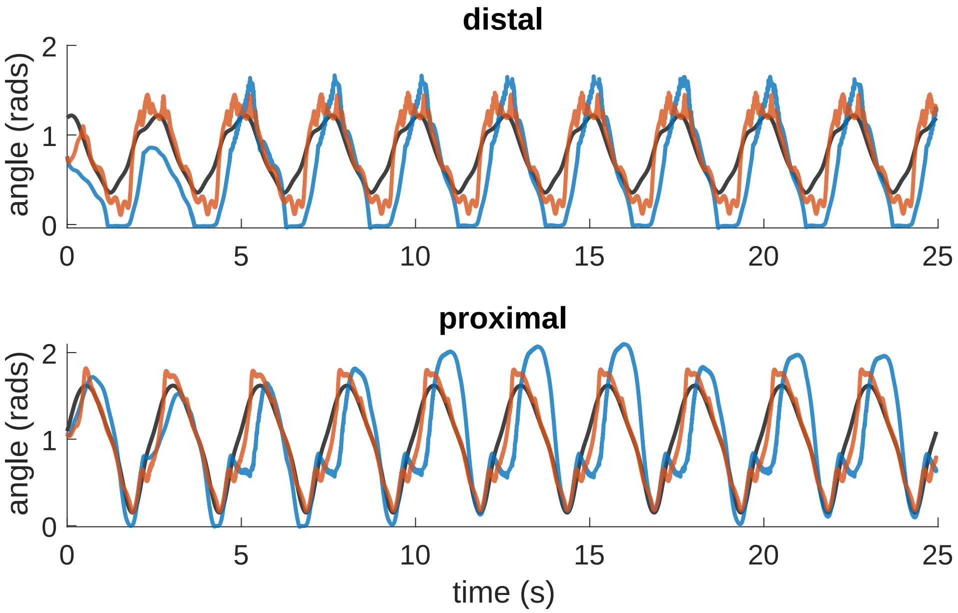


Figure 5.8: The desired (black), open-loop (blue), and the closed-loop (orange) joint angles for one sample run of the locomotion with the gantry task.

allow—for the deviation from the desired posture and maintains the prescribed posture which is standing vertically (see Supplementary Video).

5.4.5 Learning from each experience task

Biological systems subject to Hebbian learning reinforce or attenuate synaptic connections with each experience [44, 40]. Similarly, the G2P algorithm adds the input-output data from each run (i.e., experience) to its database and recalculates (i.e., refines) the inverse map with all available data before the next run (i.e., warm start of the ANN). Fig. 5.9 shows the mean and standard deviation of error over 50 random cyclical movement tasks as a function of refinement number after each experience for both the open-loop (blue) and the closed-loop (orange) systems. Both systems exhibit the expected reduction of error with increasing experience. However, this trend is accelerated in the closed-loop system where both the mean and the standard deviation of the error plateau after only 6 refinements.

We believe that the more relevant data collected by the closed-loop system contributes to this. To test this idea, after each refinement, we tested both systems with switched inverse maps. This will distinguish contribution of the error correction of the feedback signal from the potential contribution of a more precise inverse map. Open-loop system shows accelerated learning and smaller error when using the inverse map trained by the closed-loop data (green). Also, although the error for the closed-loop system with either inverse model is very small and plateaus fast (after ~ 5 refinements), it has smaller mean and standard deviation with the inverse map trained with data collected by the closed-loop system. The p-value between the 50 trials of the last refinement of closed-loop systems using closed-loop and open-loop inverse maps (orange and red curves, respectively) was 3.0927e-04. This measure for the open-loop systems (green and blue curves) was 1.0234e-07. These results show that not only does closed-loop system reduce error by commanding correction signals, but it also enhances the refinements of the inverse map by providing more task specific data at each attempt.

To demonstrate that the system will not suffer from overfitting with the proposed method and will allow generalization at the same time with improving the inverse map on-the-go, we also tested refinements for both systems for 50 random cyclical movements introduced back to back and saw similar descending trend in the error even for the movement cycles that were not experienced before. These on-the-go improvements of the inverse map are consistent with the physical system results (without real-time feedback) that we have previously reported in [90]. Note that the babbling data are always included in the refinements (as the original version of G2P [90] to make sure it will not overfit to experience alone.

5.4.6 Variable feedback delay task

Fig. 5.10 shows error over 50 random cyclical movements as a function of increasing feedback delay from 5–100 ms. We plot open-loop error (red wireframe styled lines) for the same tasks as a reference and see that the closed-loop system outperforms it for delays up to 100 ms. At very long delays, naturally, the closed-loop system will treat corrections as perturbations, its performance will degrade, and instabilities will likely arise.

5.4.7 Sensitivity to proportional-and-integral (PI) feedback gains

The choice of PI gains is traditionally made by either trial-and-error, Bode plots and, more recently, by using search algorithms (e.g. evolutionary algorithms [34]). The choice of optimal PI gains is beyond the scope of this paper. However, we briefly explored the sensitivity to a wide range of PI gains over 50 cyclical movements and found that it still yields satisfactory performance (see Supplementary Information section in

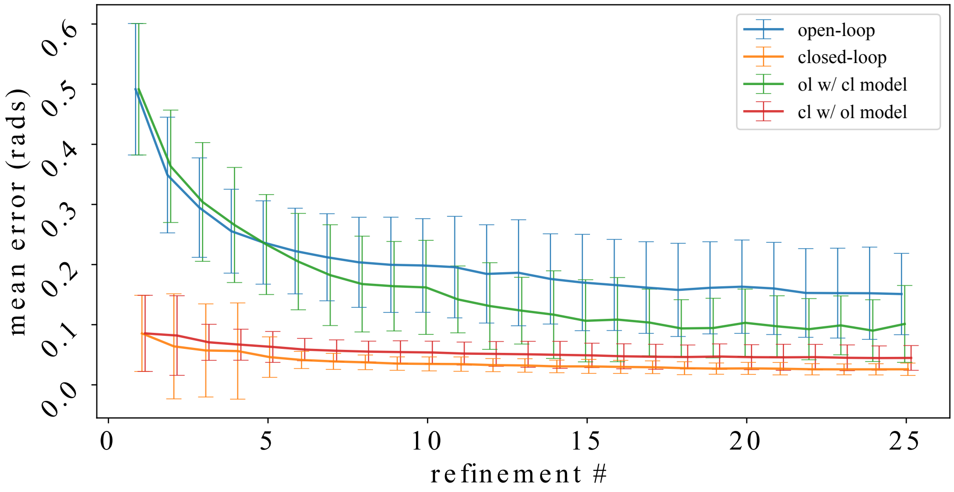


Figure 5.9: Error values for 50 random cyclical movements as a function of repetitions (and consecutive refinements) for open-loop (blue) and closed-loop (orange) systems as well as open-loop and closed-loop with switched inverse maps (green and red, respectively).

[91]—albeit with the expected faster rise times and greater overshoot with higher gains, and vice versa with lower gains.

5.5 Summary of contributions

Our method improves upon the current work in autonomous control of tendon-driven systems [90] because i) it uses the inverse map of the tendon-driven system it autonomously learned during an initial motor babbling phase and only relies on feedback to compensate for inaccuracies as needed; and, more importantly, ii) shows that by collecting more relevant data during a performance, simple feedback also facilitates and accelerates autonomous learning; naturally, more relevant experience is more useful.

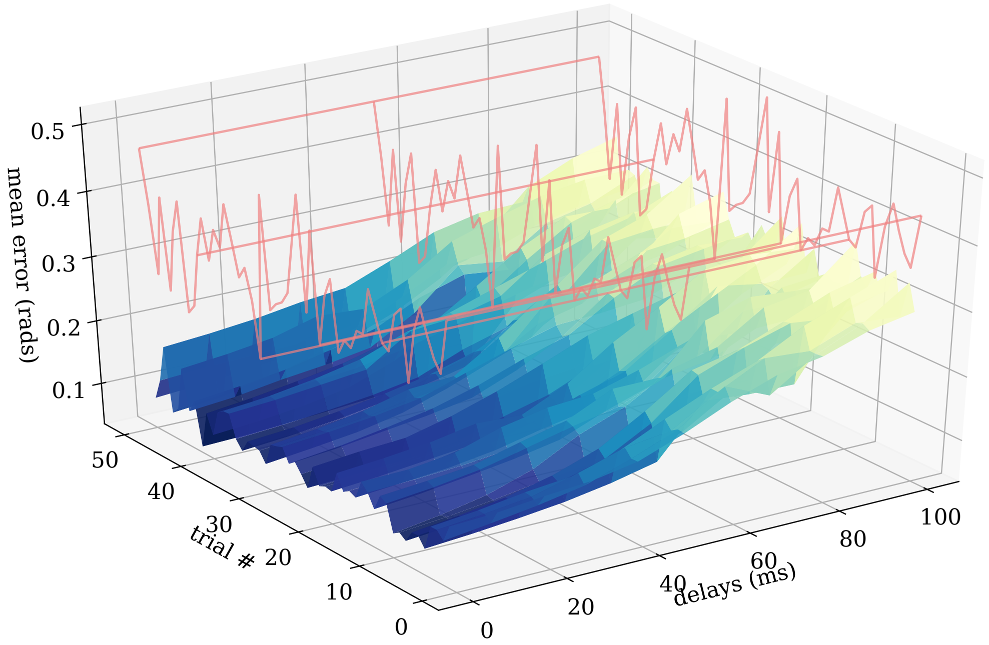


Figure 5.10: Error values for 50 random cyclical movements as a function of feedback delay for the closed-loop system. Error for the open-loop system is also provided (red wireframe styled lines) for comparison.

5.6 Discussion

We chose to apply corrections to the input velocity signal, as opposed to the input position signal, since velocity has a direction (is a vector) and can move the joint to the right position even with an imperfect model (as can be seen from the point-to-point experiment results, the outputs of position input can have high errors). Also, please remember that we chose position as the output to define error on and it is very common in controls to use the derivative of the tracked signal for correction (Air conditioning, Cruise control, etc.). However, based on the need, user can choose any other error signal (or a weighted combination) and use PID gains to feed it to the most pertinent input to entertain the correction signals.

In this paper, we showed the contributions of simple kinematic feedback in improving both performance and learning rate of the inverse map generated using limited experience while being robust to sensory delays and choice of PI parameters. We performed our test in both simulation and physical implementations of a tendon-driven robotic limb. However, it would be very interesting to test the proposed system on more complex systems such as bipeds or quadrupeds and compare their performances in more sophisticated tasks in the future work, especially in their physical implementations.

SUPPLEMENTARY INFORMATION

The code and the supplementary files (Videos, MuJoCo models, etc.) can be accessed through project's Github repository at:

https://github.com/marjanin/G2P_with_fb

ACKNOWLEDGMENTS

This project was supported by NIH Grants R01-052345 and R01-050520, award MR150091 by DoD, and award W911NF1820264 by DARPA-L2M program. Also, by USC Provost Fellowship to A.M. and the Consejo Nacional de Ciencia y Tecnología (Mexico) fellowship to D.U.-M.

Chapter 6

Model-agnostic Bio-inspired Autonomous Lifelong-learning of Kinematic Control in Tendon-driven Quadruped Robots

Ali Marjaninejad[1] and Francisco J. Valero-Cuevas[1,2]

[1]Department of Biomedical Engineering, University of Southern California, Los Angeles, CA

[2]Division of Biokinesiology and Physical Therapy, University of Southern California, Los Angeles, CA

6.1 Chapter summary

Robots will achieve true autonomy in controlling their bodies and interacting with their environments only when they are able to learn within the constraints of their designs or environments with minimal to no prior information and can learn from and adapt to

new tasks and conditions without experiencing catastrophic forgetting of prior ones. The autonomous learning of control is of an especial important in the case of tendon-driven robots, which offer a wide set of advantages such as flexibility in the placement of actuators, but also introduce further challenges to its modeling and control. Here, we have extended our bio-inspired learning algorithm, General-to-Particular, to a quadruped system and have shown its aptitude in learning to control the system within only a few seconds and performing functional movements with continuous learning. Most importantly, we have successfully utilized simple tactile sensory information to enable the system to distinguish between changes on the amount of load it carries and achieve lifelong-learning without catastrophic forgetting (continually learn without overwriting the previous skills).

6.1.0.1 Author Contribution

A. M. has designed experiments, wrote the code, run simulations and written the first draft of the manuscript. F.V-C. has provided guidance and feedback to the manuscript and during each step of the experiment design and interpreting the results.

6.2 Introduction

Vertebrates can develop motor skills from limited exposure to a task, learn from and adapt to changes when facing a new experience, generalize basic principles across different tasks, and learn how to perform new tasks without overwriting the old ones. This lifelong-learning (L2M) without catastrophic forgetting ability has equipped them with their enviable learning speed, efficiency, and adaptability even without a comprehensive prior about or model of the task, their body, or the environment. The level to which an agent with artificial intelligence can mimic these abilities will be a decisive factor in determining if it can learn, perform, and adapt in real-world applications with limited observability, incomplete or even inaccurate priors, and uncertainties in interacting with the environment or other agents.

Vertebrate do not have explicit access to their biomechanical models or its parameters (such as maximal muscle forces or the moment-arm values). Although not very often in current commercial robots, it can be the case in some of the robotic applications as well. It is an important challenge when the robot is expected to adapt to the changes it is introduced to or if it want to learn in a design-agnostic way, which opens up new avenues toward brain-body coevolution in robots. By enabling robots to learn by exploring and to create self-awareness of their body dynamics, they will be able to adapt to changes in their body dynamics or even completely new body structures.

Tendon-driven systems are great study cases for design-agnostic autonomous learning of controls since these systems are notorious for their inherent challenges in their controls. Their simultaneously over- and under-determined nature greatly constraints the feasible kinematic state space. Moreover, this makes the control problem even more challenging since there is no one-to-one relationship between the DoFs and actuators as is the case in the joint-driven systems.

Even when a successful control strategy is found for a particular task, it is important to be able to distinguish different tasks and utilize the knowledge gathered during learning one to have a head-start in learning the others. Tactile sensory information is an important type of feedback signal which can play an important part in discriminating different tasks or different phases in a single task. It has shown that tactile sensory information can provide important information such as success in a reaching task [135] or hitting the floor signal (heel strike) during locomotion [46]. Moreover, if placed on the extremities, it can be a good indicator of the weight that the limb is carrying at each moment and therefore better guide the control strategy toward adjustments on the muscle forces.

Here we have implemented and expanded our G2P algorithm to a quadruped robot with two DoFs controlled by three tendons on each limb and have studied the effects of different architectures of Artificial Neural Networks (ANNs), incorporating position error feedback, and endpoint tactile sensory information to controls and adaptation (both forward—generalizing to new tasks—, and backward—no catastrophic forgetting of old tasks) across 4 different test cases: performing limb movements in the air, on the floor, on the floor while carrying a light weight, and on the floor carrying a heavy weight. Moreover, we have tested two different curricula with the forward and reverse order of the test cares mentioned above (light to heavy and vice versa). Our results show that more specialized ANN architecture (one ANN for each limb) outperforms a single large ANN that maps all the input kinematics to the estimated muscle activations. Moreover, our system can achieve reasonable performance (RMSE of joint angles in a 10 seconds point-to-point and cyclical movement tasks) within only minutes of exploration and in the absence of any prior model of the system or the environment. Moreover, this proposed model model-agnostic approach will enable robots to learn without being limited to a specific physical design which will also open the doors to the brain-body coevolution idea (lifelong adaptation of both controls and physical structure) in robots.

6.3 Methods

Here we discuss the design of the quadruped system, tasks it is tested on the learning pipeline and the variables studied: the effects of position error feedback, tactile sensing, and the architecture of the ANNs used.

6.3.1 Quadruped design and the simulation environment

We have designed a tendon-driven quadruped with three tendons and two DoF on each limb (each limbs follows a similar design as in[90, 91]). Each tendon is actuated using a MuJoCo Muscle actuator with peak active force equal to 120N. Also, this quadruped design is an expansion of the OpenAI's HalfCheetah (which has only 2 limbs) design and inherits most of the other physical parameters from it. The resulting design file (in .xml format; more details and the xml design files are available online as a part of the supplementary files) was run by the MuJoCo physics simulator [146]. We have used 2.5 ms time step to reduce the effects of potential integration errors during the simulation.

6.3.2 Tasks and Test cases

6.3.2.1 Tasks

We have tested all our cases with two tasks: continuous cyclical movements and point-to-point movements (similar to [90, 91]

cyclical movements During this task, both proximal and distal joints follow sinusoidal trajectories with $\pi/2$ phase difference. Also, diagonal limbs are going to be synchronized together (similar to trotting). This task consists of 10 seconds with a frequency of 1.5 Hz (15 cycles during the entire task).

point-to-point movements In point-to-point task, the position of each angle is determined randomly from the range of motion of each limb (Uniformly distributed). In this case, all four limbs are following the same desired positions. This tasks consists of 10 seconds with 10 position commands for each joint (1 second in each position).

6.3.2.2 Test cases

We have tested both task in four different cases as described here.

Suspended in Air In this case, the quadruped is suspended in air and therefore limbs will not have contact with floor. In this case, the only thing the system needs to learn is how to deal with and control dynamics of its own limbs.

Walking on the floor In this case, the system is put on the floor, the limbs can interact with the floor, and therefore can move. In this case, system is going to experience contact dynamics and need to deal with the weight. The density (volumetric mass density) of all the parts of the body is set to $1000Kg/m^3$ (except the density of head, neck and tail, which are mainly added for visual purposed, and are set to $10Kg/m^3$)

Walking on the floor with a load (light) This case is similar to the walking on the floor with a slight difference that it now has to carry a load (as seen on 6.1 c.). This will have a direct effect on the weight and therefore a robust and successful control strategy would require the system to adopt to the changes caused by it. The density of the load in this case is also set to $1000Kg/m^3$

Walking on the floor with a load (heavy) In this case, the weight of the load is twice as much of the previous case; in other words, this case is identical to the previous in all aspects except the load density which is set to $2000Kg/m^3$.

6.3.3 Learning pipeline (G2P algorithm)

To be able to find a mapping between the desired kinematics and the muscle activations that would lead to them on our tendon-driven system, we use the G2P algorithm [90] which is consisted of two main parts:

6.3.3.1 Motor babbling

During this phase, muscle are randomly activated (uniformly distributed from the 0 to 100% activation range) and the resulting sensory input information (Tactile sensory information which is endpoint force values and Kinematics which consist of joint angles, angular velocities, and angular accelerations) are collected. These sensory information and activations then are going to be used to train an ANN as input and desired outputs, respectively. The resulting ANN is then going to be used to predict muscle activations required to perform the desired tasks during the refinements phase. For all cases in this study, we have performed the babbling phase for 60 seconds.

6.3.3.2 Refinements

During this phase, the kinematics of the desired task (here, either cyclical or point-to-point) are sent to the ANN to estimate the required muscle activations and next, to perform the move using those activations. Also, for the tactile sensory, we always feed the tactile sensory of the previous time step (except the first simulation step where we feed 0 on the tactile sensory input). The resulting task specific sensory information and muscle activations are concatenated with the data available so far and used to re-tune the ANN. Please note that the motor babbling provide sparse sampling within a vast volume of the sensory information while refinements enables sampling more specific to the sensory space of a desired task.

6.3.4 Scaling sensory data

Scaling and normalization of input data can enhance the learning speed and therefore improve on the data-efficiency of a machine learning algorithm. This is even more pronounced when inputs have different units and therefore can have vast differences in their ranges. To address this problem, we have scaled the input data (dividing them by their expected variance). To make sure that we are not depending on prior information, we calculate these scaling factors by running a 60 second babbling (which is done only once for the entire curriculum of tasks and test cases) when the quadruped is on the floor (no load) and use the collected data to calculate scaling factors for all cases. It is obvious that these values do not guarantee unit standard deviation for all sensory data collected across different tasks and test cases, however, it greatly reduces the differences in ranges and therefore helps in faster convergence of the ANN training processes.

6.3.5 ANN architectures

We have studied two different ANN architectures to assess which one can lead to more accurate control of movements. A single ANN for the entire system and Multiple ANNs (one for each limb). Regardless of this architecture, all ANNs start with a babbling (in air), and once they got trained, they will concatinate any data coming from new babblings or refinements and re-tune their weight using (train on) this cumulative concatenated data set while warm-starting using the weights from the last case (the ANN weights after the last training and before the addition of the new data). The error defined to train the

ANNs is the mean square error (mse) over all muscle activation (fed into the ANN vs. the estimated ones). We use MLP ANNs with one hidden layer (with 24 hidden layer neurons for the single ANN and 6 hidden layer neurons for each ANN in the multiple ANN case), linear activation functions (which we observed to perform better than the case of using sigmoid functions), and the ADAM optimizer [63] implementation in the Keras API from the Tensowflow library.

6.3.5.1 Single ANN

In this case, we use a single ANN that maps all the sensory input into the predicted muscle activations of all 12 muscles. One hypothesis is that since the ANN has access to all sensory inputs (kinematics and tactile sensory of all limbs) at the same time, it might be able to utilize these extra information to better implement the inverse dynamics and therefore enable better control of the limbs.

6.3.5.2 Multiple ANNs

In this case, we use a single ANN for each limb (that is four identical ANNs in total) that maps all the sensory input of that limb into the predicted muscle activation values (3 muscles). An opposing hypothesis to the one brought up in the Single ANN subsection is that a more focused input and output might mean fewer distractive data points for the ANN and therefore lead to better convergence and a superior control of the limbs compared to the previous case.

6.3.6 Tactile sensory information utilization

We have studied the potential contribution of tactile sensory in improving the control performance in our G2P framework; especially across different test cases (in terms of both forward and backward generalizability). A MuJoCo touch sensor (sensing the magnitude of the applied force) is used at the end of each limb (see the green areas on 6.1). We study the performance of the system across tasks and test cases with and without access to these tactile information.

6.3.7 position error feedback

Similar to [91], we have implemented corrective position error feedback on the position error of the joints and here we have studied the amount in which it contributes to the improvement of the accuracy of the limb control in our simulated quadruped across task, test cases, and ANN architectures.

6.3.8 Performance metrics

Here, we have assessed the control error on joint positions in the proposed L2M framework. As an essential part of L2M without catastrophic forgetting, the agent should demonstrate both forward learning (being able to learn as it is introduced to new tasks) and backward generalizability (being able to still perform well on the task it has learn earlier even after being trained on the newer tasks) capabilities.

6.3.8.1 Defining of error

Also, to measure how well a system has learned to perform the desired movements, we calculate the round mean square error (rmse) over the joint angles (across all limbs) for the last half of the data (to make sure the effects of transient initial conditions are washed out) and report it. We use rmse as opposed to mse since it preserves the the units of the inputs (radians).

6.3.8.2 Forward learning plots

In these series of plots, we show the error of each task and the progress of error after each babbling or refinement. Please note that for each point in these plots, we use the updated model that was trained with the cumulative data so far.

6.3.8.3 Backward generalization plot

Once babbling and refinement phases across all test cases are over and the ANN parameters are re-tuned using the most updated cumulative data set, we fix the model (ANN parameters) and test all task cases again with this final model. This will show how well the current model can generalize across all tasks it has learned so far without being re-tuned.

6.4 Results

Here, we have demonstrated the effect of studied configurations on the control accuracy over the joint angles. At each figure, we have only changed the configuration of the interest (position error feedback, tactile sensory information, and the ANN structure) and compared the results while the other configurations are kept the same.

6.4.1 Effects of position error feedback

Fig. 6.1 shows the forward learning plots for the configuration with and without position error feedback in orange and blue, respectively. Also, Fig. 6.1a-d are corresponding to the test cases of: in air, on the floor, on the floor with the light load, and on the floor with the heavy load, respectively Also, fig. 6.2 shows the backward generalization plot for all the test cases mentioned above for the configuration with and without position error feedback (for each test case, right and left box plots, respectively).

As you can see, the plots on Fig. 6.1 show that the having position error feedback greatly reduces the time needed for the error curves to converge and therefore reduce the number of refinements needed for the error plots to flatten. Also, Fig. 6.2 (backward generalization results) show that the corrective position error feedback significantly reduces the joint position error across all test cases (One and two stars represent p-valeus equal or smaller than 0.05 and 0.01, respectively, for a one way ANOVA analysis).

In terms of other configurations, these plots represent the results where there is no tactile feedback and the system uses multiple ANNs, however, the general pattern is

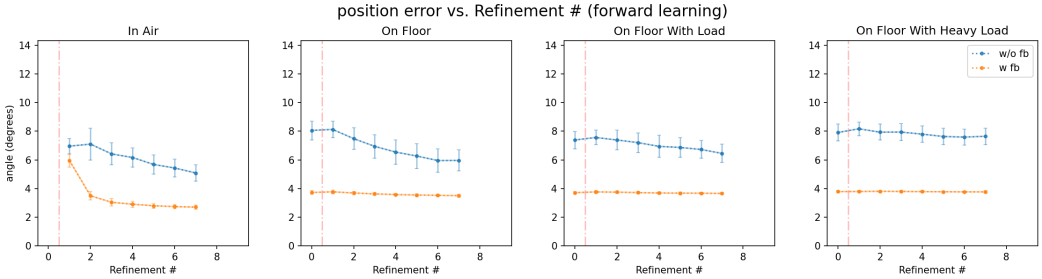


Figure 6.1: Forward learning plots across test cases with (orange) and without (blue) position error feedback.

consistent across other variations (position error feedback always enhances the performance). Please see the supplementary documents for the complete set of result plots for all possible configurations.

6.4.2 Effects of ANN architecture

Fig. 6.3a-d show the forward learning plots for for all cases for the configuration with multiple, and single ANNs in orange and blue, respectively. Fig. 6.4 shows the backward generalization plot for all the test cases for the configuration with a multiple and single ANNs (for each test case, right and left box plots, respectively).

As can be seen on both figures, the multiple ANN structure outperforms the single one. Moreover, this difference in performance is more significant in tasks that are experiencing the effects of weight which suggests that multiple ANN structure can better handle (is more robust) the unexpected dynamics of interactions with the floor that are caused by the weight.

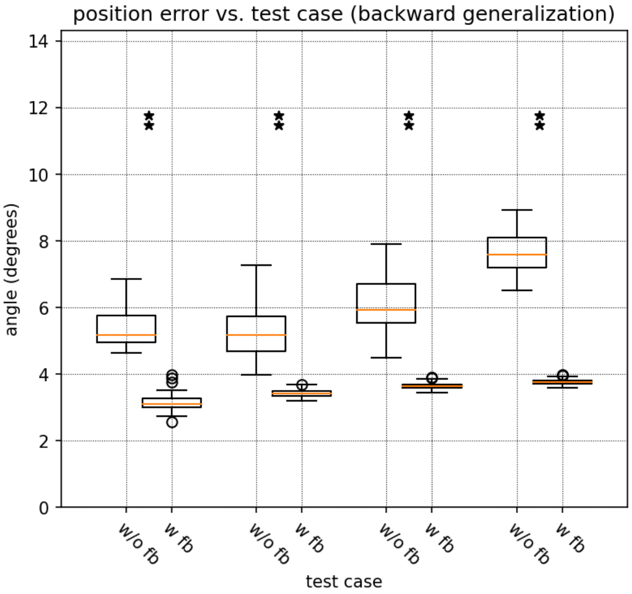


Figure 6.2: Generalization plots across test cases with (right boxplot for each case) and without (left boxplot for each case) position error feedback.

One hypothesis would be that since the single ANN has more coefficients (weights and biases) to set, it would need more data to converge. However, we saw a similar pattern (multiple ANNs outperforming the single ANN) even when trained on larger data set. Moreover, the fact that the forward learning curves are almost flat after the second case rejects this hypothesis.

Alternatively, a larger ANN might be more likely to suffer from local minima. In addition, the kinematic input from other limbs seems not to be very helpful and therefore create destructive interference. The differences in the performance are even more statistically significant when we enable position error feedback (see supplementary information). This reinforces the destructive interference in the single ANN hypothesis since there is only a single corrective position error feedback signal for each joint for the

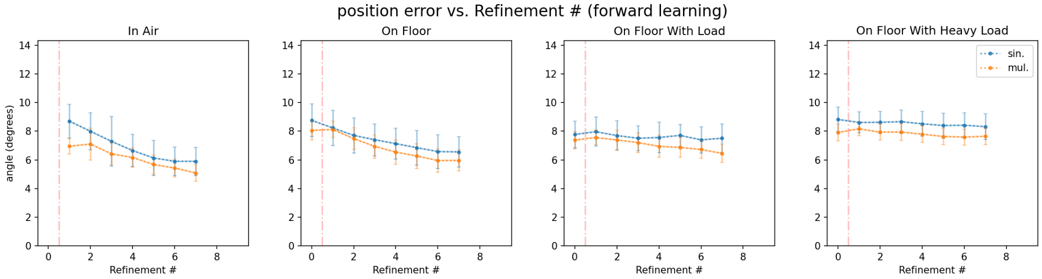


Figure 6.3: (Forward learning plots across test cases with multiple (orange) and single (blue) ANN structure.

multiple ANN structure being fed back to the ANN while all the corrective position error feedback signals are fed back to the same ANN in the single ANN structure.

6.4.3 Effects of tactile sensory information

Fig. 6.5a-d show the forward learning plots for for all cases for the configuration with and without tactile sensory in orange and blue, respectively. Fig. 6.6 shows the backward generalization plot for all the test cases for the configuration with and without tactile sensory (for each test case, right and left box plots, respectively).

Here, we have selected the results with configuration without position error feedback to isolate the contributions of tactile sensory. We see that pretense of feedback highly enhances the accuracy and therefore covers most of the contributions of the tactile sensory (please see the supplementary figures).

As can be seen on both figures, overall, the tactile feedback contributes in enhanced performance. It is especially the case for the backward generalization results of the more extreme cases (on air and with the heavy load). We believe that it is because the presence

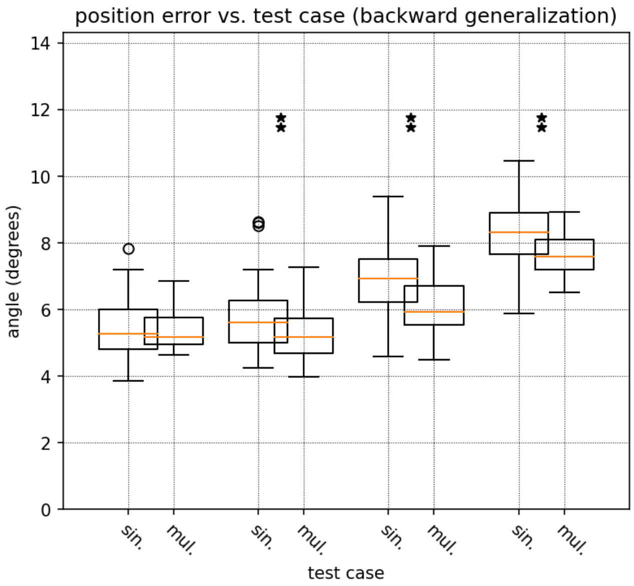


Figure 6.4: Generalization plots across test cases with multiple (right boxplot for each case) and single (left boxplot for each case) ANN structure.

of the tactile sensory enables the mapping to differentiate cases with different weight and therefore select the appropriate muscle activation values for each case. On the contrary, the configuration without the tactile sensory do not have access to these discriminatory information and the ANN tries to minimize the error over the entire data set (across all cases) and therefore would not have major performance drop for the ”on floor” and ”with light load” cases but would suffer more (compare to the configuration with the tactile sensory) on the more extreme cases (”in air” and ”with heavy load”).

Importantly, we see an even more pronounced contribution for the sensory in the single ANN structure (please see the supplementary info) that is persistent for all four test cases. It suggests that having access to sensory signals from other limbs can further enhance the performance since it provides a more complete set of information regarding

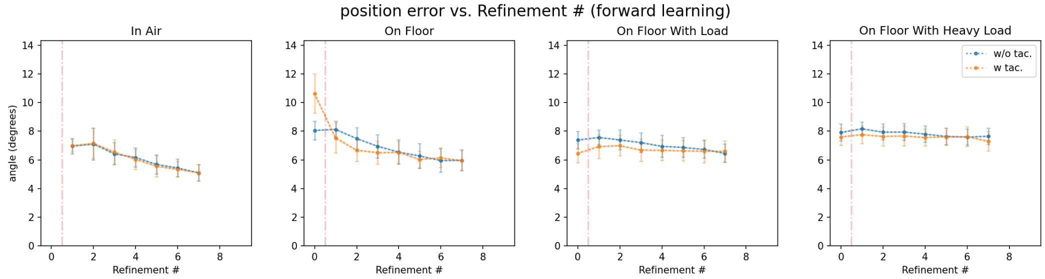


Figure 6.5: Forward learning plots across test cases with (orange) and without (blue) tactile sensory information.

the total weight of the system and the part that is being carried by each leg. Therefore, similar to biological systems, it seems like having specialized networks that are interconnected with each other at some level (here, sharing the sensory information of each limb with other limbs) seems to be a promising future design architecture to pursue.

Also, please note that the ”with tactile sensory” case has a relatively higher error on the 0^{th} refinement of the ”on the floor” case (the test results before babbling or being trained on the information collected from the new test case). This is because the input nodes associated with the tactile sensory information so far have always fed with zeros (since there is no touch during the ”in air” case) and therefore, the weights associated with them are not refined. These weights, however, are pruned right after the system is exposed to the new environment and the ANNs are trained with data including tactile sensory information.

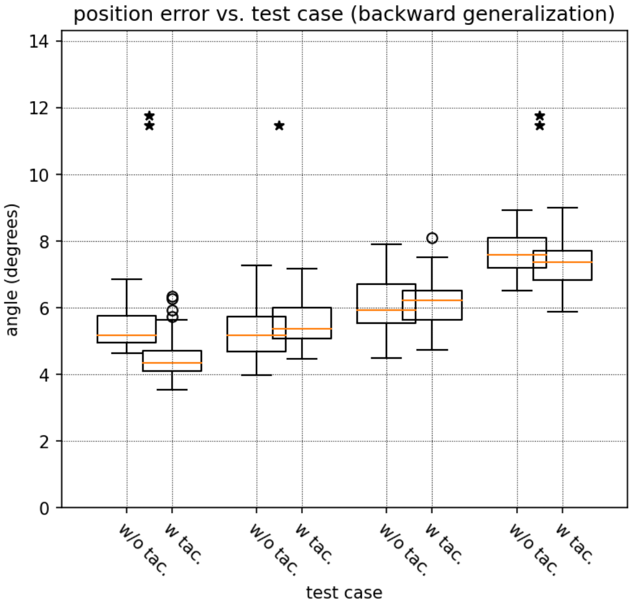


Figure 6.6: Generalization plots across test cases with (right boxplot for each case) and without (left boxplot for each case) tactile sensory information.

6.4.4 Stacking all improvements

In this part, in order to assess the overall contribution of these configurations, we have stacked all the configurations that had lead to an improvement in performance (A system with position error feedback, tactile sensory, and multiple ANNs) and have compared the resulting performance with the system without any of these configurations. We label these two systems as higher performance and lower performance configurations, respectively. Please note that the without enhancement system is similar in configurations to the original G2P algorithm presented in [90].

Fig. 6.7 and 6.8 shows the plots for the forward learning and backward generalization results, respectively. As it is clear from these figures, there is a significant improvement in the control performance for the higher performance configurations in both forward

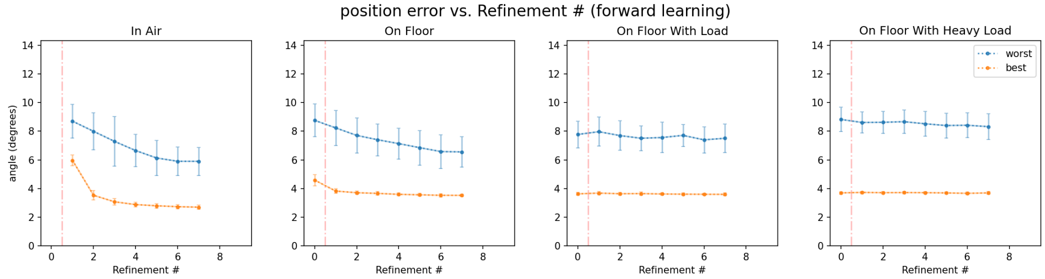


Figure 6.7: Forward learning plots across test cases with all enhancing configurations stacked together (orange) and without them (blue).

learning and backward generalization which leads to a better L2M without catastrophic forgetting performance in achieving kinematic control which is the main focus of this paper.

6.5 Conclusion

Here we have implemented task-agnostic autonomous learning for a bio-inspired tendon-driven quadruped and have studied the effects of different configurations on its kinematic control performance. Our results show that the proposed framework is able to learn without any prior knowledge about the design parameters just after a minute of random kinematic exploration (motor babbling) and a few attempts of the task of interest (followed by refinements).

We have shown that the corrective position error feedback that utilized the forward model generated using the G2P algorithm significantly improves the performance. Also, our results show that smaller and more specific neural networks would outperform a single all-in all-out network when only dealt with kinematic information. However, our

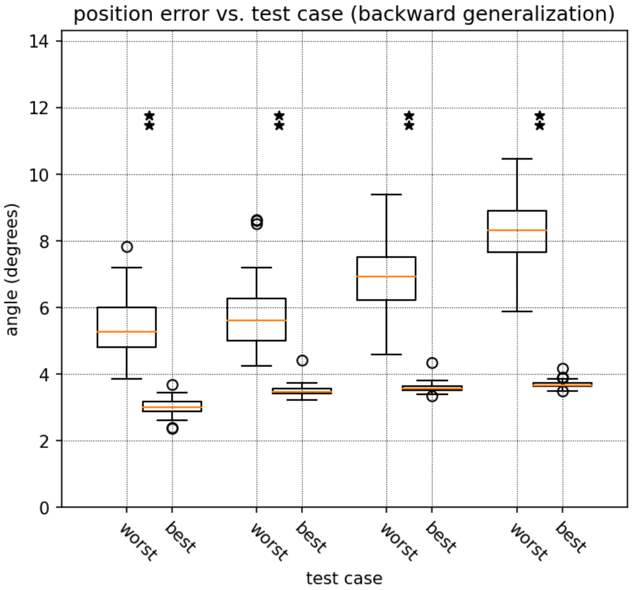


Figure 6.8: Generalization plots across test cases with all enhancing configurations stacked together (right boxplot for each case) and without them (left boxplot for each case).

results suggest that having access to tactile sensory information from other limbs can further enhance the performance of each limbs.

Our proposed framework was able to achieve RMSE of less than 4° in controlling joint positions of a tendon-driven quadruped which satisfied the main goal of this research. Moreover, we have shed some light into the effect of contributions of ANN structure and tactile sensory in kinematic control which can help future work in designing even more accurate and data-efficient control frameworks.

6.6 Limitations and future work

As brought up in the previous section, our results suggest that although a single ANN for each limb in general outperformed an all-in all-out approach, having access to tactile information of other limbs can favorably affect the kinematic control of a limb. This makes sense since by having access to these information, an ANN would be able to make a better assumption regarding the total weight of the system what fraction of it that the limbs needs to taken care. This is also supported by evidence seen in biology where smaller and densely connected local networks are more sparsely interconnected to other networks at some levels [11, 88]. Implementation of such a hybrid ANN architecture is beyond the scope of the current work but is a very interesting avenue to pursue for future studies.

Here we have achieved to satisfactory levels of kinematic control and it would be interesting to attach it to a higher level controlled (in a hierarchical fashion) to perform functional tasks. Moreover, here we have only shown the effect of studied configurations in kinematic control; however, it would be interesting to also study contributions of the sensory information (either tactile or kinematic) on higher level planing and control task. For example, sensory signals are known to contribute significantly in high level task planning and dexterous manipulation tasks in biological systems [35, 151].

Moreover, although we have used a physically faithful simulator (MuJoCo), it would be an interesting future work to also assess the performance of the system in a real-world, physical implementation. Lastly, here we have studied the L2M capabilities of the proposed framework in adapting and generalizing to different weight loads, however, assessing the adaptiveness of such L2M algorithms to changes to the system design (wear and tear, partial damage or loss of a limb, etc.) would be another interesting research direction that will further pave the way for highly robust, adaptive, and multipurpose L2M robots.

SUPPLEMENTARY INFORMATION

The code and the supplementary files (Videos, MuJoCo models, etc.) can be accessed through project's Github repository at:
https://github.com/marjanin/quadruped

ACKNOWLEDGMENTS

This project was supported by NIH Grants R01-052345 and R01-050520, award MR150091 by DoD, and award W911NF1820264 by DARPA-L2M program.

Chapter 7

Conclusion and future direction

Robots can only become useful across multiple real-world tasks and scenarios when they can learn from limited experience and adapt to new and unexpected scenarios. Moreover, model-agnostic autonomous learning would further enhance their practicality by reducing the amount of prior knowledge needed to control their bodies and perform functional tasks. This is especially important in tendon-driven systems (or non-rigid systems in general) that are known to be difficult to model or control, yet have a lot to offer such as a large set of feasible output kinematics and flexibility in actuator placement.

In this project, I have built a framework in which robots can autonomously learn to control their tendon-driven structures within minutes of exploration, and in a model- and task-agnostic fashion.

In Chapter 2, we elaborated on tendon-driven systems, their underlying principles, challenges in their control, and the benefits they can provide. In Chapter 3, we introduced the G2P algorithm which enables autonomous learning of functional tasks in robots from limited experience and tested it on a single tendon-driven leg connected to a gantry. In

Chapter 4, we tested the abilities of G2P to adapt and utilize added dynamical complexities to the leg design in Chapter 3 in terms of adding parallel stiffness to the tendon actuators. We showed that not only can it handle these added dynamical complexities but also it can utilize them to accelerate learning and enhance performance. In Chapter 5, we implemented a corrective position feedback controller which complements G2P's feedforward control and showed how it can significantly improve performance across cyclical or point-to-point tasks in both simulation and hardware implementations. Lastly, in Chapter 6, we studied different strategies in implementing an accurate and robust G2P based control strategy for a complete quadruped and showcase precise kinematic control without catastrophic forgetting for locomotion.

I believe that this work represents a significant advancement in the field of biologically-inspired and algorithmically-rigorous autonomous learning in robots, and minimizes their dependencies on their physical designs. This will enable the emergence of robots that can learn on-the-fly and adapt to real-world situations in unstructured environments. Moreover, the fact that our systems are able to work with non-rigid structures, and especially tendon-driven architectures, enables autonomous robots to enjoy new levels of agility, energy efficiency, and safety. This algorithm and hardware (brain-body) combination is a key toward a new generation of robots that can contribute to a wider range of non-routine applications in unstructured environments from post-disaster rescue missions to practical everyday tasks, to planetary exploration.

It is important to note that the proposed versions of G2P can also be used as a kinematic learning and control module along with other learning algorithms and hierarchical/distributed control architectures. This means that it can be used in a hierarchical way to enhance learning by bypassing the inverse kinematic mapping in end-to-end learning algorithms such as Proximal Policy Optimization (PPO). It can also be the ‘predictive step’ in Model Predictive Control (MPC) algorithms. Moreover, although the proposed algorithm is model-agnostic, I encourage the use of a model (even if preliminary) to speed up the process of learning whenever such a model is available. One simple way of using a model to enhance (i.e., ‘warm-start’) learning would be to use the model to run the forward dynamics and generate pairs of actuation-to-kinematics data that can then be fed into the algorithm to train it and provide it with a warm-start. Finally, the proposed G2P approach can be used not only to predict inverse dynamics but also to randomly explore and model other problems. For example, more recently, we have shown its utility as an agnostic observer that predicts joint angles from biologically-tenable sensory signals in mammalian muscles [42].

In terms of the future work, I believe that the field can and should benefit from modular structures in both hardware design and learning algorithms. Modular and hierarchical learning and control algorithms can further enhance data-efficiency by breaking complicated tasks into smaller and more specific sub-tasks, to then tackle them individually. Modular design can provide robots with a wider range of applications across multiple body architectures to reduce the production costs while increasing the range of applications. Moreover, transfer learning across tasks and across agents would be an important topic to focus on. Inheriting basic principles learned by one robots and shared with others across a number of tasks can further enhance data-efficiency by providing all robots with a warm-start during the learning process of new tasks. Meanwhile, inter-agent share of information in multi-agent cases can help with parallel acquisition of data and faster learning of skills, especially on the higher-level side of tasks that are (or should be) agnostic to kinematics or design details of individual robots, environments and tasks. Lastly, current research on the field of bio-inspired robotics can not only provide inspiration from biology for development of new and enhanced robotic structures and algorithms. This research can also lead to testbeds to 'put biological theories to the ultimate test of physical implementation' and answer questions in precise, biologically-faithful and well-structured robotic implementations.

Reference List

[1] Karen E Adolph, Whitney G Cole, Meghana Komati, Jessie S Garciaguirre, Daryaneh Badaly, Jesse M Lingeman, Gladys LY Chan, and Rachel B Sotsky. How do you learn to walk? thousands of steps and dozens of falls per day. *Psychological science*, 23(11):1387–1394, 2012.

[2] Arvind Ananthanarayanan, Mojtaba Azadi, and Sangbae Kim. Towards a bio-inspired leg design for high-speed running. *Bioinspiration & biomimetics*, 7(4):046005, 2012.

[3] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*, 2018.

[4] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[5] Sarine Babikian, Francisco J Valero-Cuevas, and Eva Kanso. Slow movements of bio-inspired limbs. *Journal of Nonlinear Science*, 26(5):1293–1309, 2016.

[6] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[7] Andrea Bonarini, Alessandro Lazaric, and Marcello Restelli. Incremental skill acquisition for self-motivated learning animats. In *International Conference on Simulation of Adaptive Behavior*, pages 357–368. Springer, 2006.

[8] Josh Bongard. Morphological change in machines accelerates the evolution of robust behavior. *Proceedings of the National Academy of Sciences*, 108(4):1234–1239, 2011.

[9] Josh Bongard, Victor Zykov, and Hod Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, 2006.

[10] Oliver Brock and Francisco Valero-Cuevas. Transferring synergies from neuroscience to robotics comment on "hand synergies: Integration of robotics and neuroscience for understanding the control of biological and artificial hands" by m. santello et al. *Physics of life reviews*, 17:27, 2016.

[11] Ansgar Büschges. Sensory control and organization of neural networks mediating coordination of multisegmental organs for locomotion. *Journal of neurophysiology*, 93(3):1127–1135, 2005.

[12] Manuel G Catalano, Giorgio Grioli, Edoardo Farnioli, Alessandro Serio, Cristina Piazza, and Antonio Bicchi. Adaptive synergies for the design and control of the pisa/iit softhand. *The International Journal of Robotics Research*, 33(5):768–782, 2014.

[13] Ahmet Cetinkaya and Tomohisa Hayakawa. Sampled-data delayed feedback control for stabilizing unstable periodic orbits. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 1409–1414. IEEE, 2015.

[14] Ahmet Cetinkaya, Tomohisa Hayakawa, and Mohd Amir Fikri bin Mohd Taib. Stabilizing unstable periodic orbits with delayed feedback control in act-and-wait fashion. *Systems & Control Letters*, 113:71–77, 2018.

[15] EY Chao and Kai-Nan An. Graphical interpretation of the solution to the redundant problem in biomechanics. *Journal of Biomechanical Engineering*, 100(3):159–167, 1978.

[16] Vasek Chvatal. *Linear programming*. Macmillan, 1983.

[17] Brian A Cohn, May Szedlák, Bernd Gärtner, and Francisco J Valero-Cuevas. Feasibility theory reconciles and informs alternative approaches to neuromuscular control. *Frontiers in computational neuroscience*, 12, 2018.

[18] Steven H Collins, M Bruce Wiggin, and Gregory S Sawicki. Reducing the energy cost of human walking using an unpowered exoskeleton. *Nature*, 522(7555):212–215, 2015.

[19] Roy D Crowninshield and Richard A Brand. A physiologically based criterion of muscle force prediction in locomotion. *Journal of biomechanics*, 14(11):793–801, 1981.

[20] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503, 2015.

[21] Raphael Deimel and Oliver Brock. A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research*, 35(1-3):161–185, 2016.

[22] Cosimo Della Santina, Dominic Lakatos, Antonio Bicchi, and Alin Albu-Schaeffer. Using nonlinear normal modes for execution of efficient cyclic motions in articulated soft robots. *arXiv preprint arXiv:1806.08389*, 2018.

[23] John C Doyle. Guaranteed margins for lqg regulators. *IEEE Transactions on automatic Control*, 23(4):756–757, 1978.

[24] Aaron D'Souza, Sethu Vijayakumar, and Stefan Schaal. Learning inverse kinematics. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 1, pages 298–303. IEEE, 2001.

[25] Nima Fazeli, Miquel Oller, Jiajun Wu, Zheng Wu, Joshua B Tenenbaum, and Alberto Rodriguez. See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion. *Science Robotics*, 4(26), 2019.

[26] Nima Fazeli, Samuel Zapolsky, Evan Drumwright, and Alberto Rodriguez. Learning data-efficient rigid-body contact models: Case study of planar impact. *arXiv preprint arXiv:1710.05947*, 2017.

[27] Denise S Feirstein, Ivan Koryakovskiy, Jens Kober, and Heike Vallery. Reinforcement learning of potential fields to achieve limit-cycle walking. *IFAC-PapersOnLine*, 49(14):113–118, 2016.

[28] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M Dollar, and Danica Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77, 2016.

[29] Michael S Fine and Kurt A Thoroughman. Trial-by-trial transformation of error into sensorimotor adaptation changes with environmental dynamics. *Journal of neurophysiology*, 98(3):1392–1404, 2007.

[30] James M Finley and Amy J Bastian. Associations between foot placement asymmetries and metabolic cost of transport in hemiparetic gait. *Neurorehabilitation and neural repair*, 31(2):168–177, 2017.

[31] Jörg KH Franke, Gregor Köhler, André Biedenkapp, and Frank Hutter. Sample-efficient automated deep reinforcement learning. *arXiv preprint arXiv:2009.01555*, 2020.

[32] Jiaxin L Fu and Nancy S Pollard. On the importance of asymmetries in grasp quality metrics for tendon driven hands. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 1068–1075. IEEE, 2006.

[33] Thomas Geijtenbeek, Michiel Van De Panne, and A Frank Van Der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics (TOG)*, 32(6):1–11, 2013.

[34] Arezou Geramipour, Mohammad Khazaei, Ali Marjaninejad, and Mehdi Khazaei. Design of fpga-based digital pid controller using xilinx sysgen® for regulating blood glucose level of type-i diabetic patients. *Int J Mechatron Electr Comput Technol*, 3(7):56–69, 2013.

[35] Hanna Gertz, Dimitris Voudouris, and Katja Fiehler. Reach-relevant somatosensory signals modulate tactile suppression. *Journal of neurophysiology*, 117(6):2262–2268, 2017.

[36] Arjan Gijsberts and Giorgio Metta. Real-time model learning using incremental sparse spectrum gaussian process regression. *Neural networks*, 41:59–69, 2013.

[37] John Gosline, Margo Lillie, Emily Carrington, Paul Guerette, Christine Ortlepp, and Ken Savage. Elastic proteins: biological roles and mechanical properties. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 357(1418):121–132, 2002.

[38] Ambarish Goswami and Prahlad Vadakkepat. *Humanoid robotics: a reference*. Springer, 2019.

[39] Sten Grillner. Locomotion in vertebrates: central mechanisms and reflex interaction. *Physiological reviews*, 55(2):247–304, 1975.

[40] Sten Grillner. Biological pattern generation: the cellular and computational logic of networks in motion. *Neuron*, 52(5):751–766, 2006.

[41] Sten Grillner and Peter Wallen. Central pattern generators for locomotion, with special reference to vertebrates. *Annual review of neuroscience*, 8(1):233–261, 1985.

[42] Daniel A Hagen, Ali Marjaninejad, and Francisco J Valero-Cuevas. A bio-inspired framework for joint angle estimation from non-collocated sensors in tendon-driven systems. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7778–7783. IEEE, 2020.

[43] Daniel A Hagen and Francisco J Valero-Cuevas. Similar movements are associated with drastically different muscle contraction velocities. *Journal of Biomechanics*, 2017.

[44] Donald Olding Hebb. *The organization of behavior: a neuropsychological theory*. Science Editions, 1962.

[45] Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, Martin Riedmiller, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

[46] Nicholas B Holowka, Bert Wynands, Tina J Drechsel, Andrew K Yegian, Victoria A Tobolsky, Paul Okutoyi, Robert Mang'eni Ojiambo, Diresibachew W Haile, Timothy K Sigei, Claudio Zippenfennig, et al. Foot callus thickness does not trade off protection for tactile sensitivity during walking. *Nature*, 571(7764):261–264, 2019.

[47] Alexander Hunt, Nicholas Szczecinski, and Roger Quinn. Development and training of a neural controller for hind leg walking in a dog robot. *Frontiers in neurorobotics*, 11:18, 2017.

[48] Jonathan W Hurst. *The role and implementation of compliance in legged locomotion*. PhD thesis, Carnegie Mellon University, The Robotics Institute, 2008.

[49] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. In *Conference on Neural Information Processing Systems*, 2002.

[50] Tony GJ Ingram, Jack P Solomon, David A Westwood, and Shaun G Boe. Movement related sensory feedback is not necessary for learning to execute a motor skill. *Behavioural brain research*, 359:135–142, 2019.

[51] Joshua M Inouye, Jason J Kutch, and Francisco J Valero-Cuevas. A novel synthesis of computational approaches enables optimization of grasp quality of tendon-driven hands. *IEEE Transactions on Robotics*, 28(4):958–966, 2012.

[52] Joshua M Inouye, Jason J Kutch, and Francisco J Valero-Cuevas. Optimizing the topology of tendon-driven fingers: Rationale, predictions and implementation. In *The Human Hand as an Inspiration for Robot Hand Development*, pages 247–266. Springer, 2014.

[53] Joshua M Inouye and Francisco J Valero-Cuevas. Anthropomorphic tendon-driven robotic hands can exceed human grasping capabilities following optimization. *The International Journal of Robotics Research*, 33(5):694–705, 2014.

[54] Joshua M Inouye and Francisco J Valero-Cuevas. Muscle synergies heavily influence the neural control of arm endpoint stiffness and energy consumption. *PLoS computational biology*, 12(2):e1004737, 2016.

[55] Sumit Jain and C Karen Liu. Controlling physics-based characters using soft contacts. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–10, 2011.

[56] Kian Jalaleddini, Akira Nagamori, Christopher M Laine, Mahsa A Golkar, Robert E Kearney, and Francisco J Valero-Cuevas. Physiological tremor increases when skeletal muscle is shortened: implications for fusimotor control. *The Journal of Physiology*, 2017.

[57] Kian Jalaleddini, Chuanxin Minos Niu, Suraj Chakravarthi Raja, Won Joon Sohn, Gerald E Loeb, Terence D Sanger, and Francisco J Valero-Cuevas. Neuromorphic meets neuromechanics, part ii: the role of fusimotor drive. *Journal of neural engineering*, 14(2):025002, 2017.

[58] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12627–12637, 2019.

[59] Roland S Johansson and Kelly J Cole. Sensory-motor coordination during grasping and manipulative actions. *Current opinion in neurobiology*, 2(6):815–823, 1992.

[60] Mitsuo Kawato, Yoji Uno, Michiaki Isobe, and Ryoji Suzuki. Hierarchical neural network model for voluntary movement with application to robotics. *IEEE Control Systems Magazine*, 8(2):8–15, 1988.

[61] Luke A Kelly, Andrew G Cresswell, and Dominic J Farris. The energetic behaviour of the human foot across a range of running speeds. *Scientific reports*, 8(1):1–6, 2018.

[62] Jonathan P King, Dominik Bauer, Cornelia Schlagenhauf, Kai-Hung Chang, Daniele Moro, Nancy Pollard, and Stelian Coros. Design. fabrication, and evaluation of tendon-driven multi-fingered foam hands. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9. IEEE, 2018.

[63] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[64] Hiroaki Kobayashi, Kazuhito Hyodo, and Daisuke Ogane. On tendon-driven robotic mechanisms with redundant tendons. *The International Journal of Robotics Research*, 17(5):561–571, 1998.

[65] Hiroaki Kobayashi and Ryuta Ozawa. Adaptive neural network control of tendon-driven mechanisms with elastic tendons. *Automatica*, 39(9):1509–1519, 2003.

[66] Taisuke Kobayashi, Kosuke Sekiyama, Yasuhisa Hasegawa, Tadayoshi Aoyama, and Toshio Fukuda. Unified bipedal gait for autonomous transition between walking and running in pursuit of energy minimization. *Robotics and Autonomous Systems*, 103:27–41, 2018.

[67] Sanjay Krishnan, Animesh Garg, Richard Liaw, Brijen Thananjeyan, Lauren Miller, Florian T Pokorny, and Ken Goldberg. Swirl: A sequential windowed

inverse reinforcement learning algorithm for robot tasks with delayed rewards. *The international journal of robotics research*, 38(2-3):126–145, 2019.

[68] Vikash Kumar, Abhishek Gupta, Emanuel Todorov, and Sergey Levine. Learning dexterous manipulation policies from experience and imitation. *arXiv preprint arXiv:1611.05095*, 2016.

[69] Vikash Kumar, Yuval Tassa, Tom Erez, and Emanuel Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6808–6815. IEEE, 2014.

[70] Manish U Kurse, Hod Lipson, and Francisco J Valero-Cuevas. Extrapolatable analytical functions for tendon excursions and moment arms from sparse datasets. *IEEE Transactions on Biomedical Engineering*, 59(6):1572–1582, 2012.

[71] Jason J Kutch and Francisco J Valero-Cuevas. Challenges and new approaches to proving the existence of muscle synergies of neural origin. *PLoS computational biology*, 8(5):e1002434, 2012.

[72] Robert Kwiatkowski and Hod Lipson. Task-agnostic self-modeling machines. *Science Robotics*, 4(26):eaau9354, 2019.

[73] Christopher M Laine, Akira Nagamori, and Francisco J Valero-Cuevas. The dynamics of voluntary force production in afferented muscle influence involuntary tremor. *Frontiers in computational neuroscience*, 10, 2016.

[74] Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. Scalable muscle-actuated human simulation and control. *ACM Transactions on Graphics (TOG)*, 38(4):73, 2019.

[75] Young-Tae Lee, Hyouk-Ryeol Choi, Wan-Kyun Chung, and Youngil Youm. Stiffness control of a coupled tendon-driven robot hand. *IEEE Control Systems*, 14(5):10–19, 1994.

[76] JNAL Leijnse. A generic morphological model of the anatomic variability in the m. flexor digitorum profundus, m. flexor pollicis longus and mm. lumbricales complex. *Cells Tissues Organs*, 160(1):62–74, 1997.

[77] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[78] Gerald E Loeb. Overcomplete musculature or underspecified tasks? *Motor control*, 4(1):81–83, 2000.

[79] Gerald E Loeb. Optimal isn't good enough. *Biological cybernetics*, 106(11):757–765, 2012.

[80] Gerald E Loeb, Frances JR Richmond, and Lucinda L Baker. The bion devices: injectable interfaces with peripheral nerves and muscles. *Neurosurgical focus*, 20(5):1–9, 2006.

[81] Kendall Lowrey, Svetoslav Kolev, Jeremy Dao, Aravind Rajeswaran, and Emanuel Todorov. Reinforcement learning for non-prehensile manipulation: Transfer from simulation to physical system. In *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 35–42. IEEE, 2018.

[82] Qi Luo, Zhuozhi Zhang, Jiayue Liu, Chih-hong Chou, Manzhao Hao, Ning Lan, and Chuanxin M Niu. Design of a biomimetic control system for tendon-driven prosthetic hand. In *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pages 528–531. IEEE, 2018.

[83] Poramate Manoonpong, Tao Geng, Tomas Kulvicius, Bernd Porr, and Florentin Wörgötter. Adaptive, fast walking in a biped robot under neuronal control and learning. *PLoS Comput Biol*, 3(7):e134, 2007.

[84] Alonso Marco, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. Automatic lqr tuning based on gaussian process global optimization. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 270–277. IEEE, 2016.

[85] Katherine L Mardula, Ravi Balasubramanian, and Christopher H Allan. Implanted passive engineering mechanism improves hand function after tendon transfer surgery: a cadaver-based study. *Hand*, 10(1):116–122, 2015.

[86] Ali Marjaninejad, Rohit Annigeri, and Francisco J Valero-Cuevas. Model-free control of movement in a tendon-driven limb via a modified genetic algorithm. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1767–1770. IEEE, 2018.

[87] Ali Marjaninejad, Jasmine A Berry, and Francisco J Valero-Cuevas. An analytical approach to posture-dependent muscle force and muscle activation patterns. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2068–2071. IEEE, 2018.

[88] Ali Marjaninejad and James M Finley. A model-based exploration of the role of pattern generating circuits during locomotor adaptation. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 21–24. IEEE, 2016.

[89] Ali Marjaninejad, Jie Tan, and Francisco Valero-Cuevas. Autonomous control of a tendon-driven robotic limb with elastic elements reveals that added elasticity

can enhance learning. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 4680–4686. IEEE, 2020.

[90] Ali Marjaninejad, Darío Urbina-Meléndez, Brian A Cohn, and Francisco J Valero-Cuevas. Autonomous functional movements in a tendon-driven limb via limited experience. *Nature machine intelligence*, 1(3):144, 2019.

[91] Ali Marjaninejad, Darío Urbina-Meléndez, and Francisco J Valero-Cuevas. Simple kinematic feedback enhances autonomous learning in bio-inspired tendon-driven systems. *arXiv preprint arXiv:1907.04539*, 2019.

[92] Ali Marjaninejad and Francisco J Valero-Cuevas. Should anthropomorphic systems be "redundant"? In *Biomechanics of Anthropomorphic Systems*, pages 7–34. Springer, 2019.

[93] Hugo Gravato Marques, Arjun Bharadwaj, and Fumiya Iida. From spontaneous motor activity to coordinated behaviour: a developmental model. *PLoS Comput Biol*, 10(7):e1003653, 2014.

[94] Anirban Mazumdar, Steven J Spencer, Clinton Hobart, Jonathan Salton, Morgan Quigley, Tingfan Wu, Sylvain Bertrand, Jerry Pratt, and Stephen P Buerger. Parallel elastic elements improve energy efficiency on the steppr bipedal walking robot. *IEEE/ASME Transactions on Mechatronics*, 22(2):898–908, 2016.

[95] Patricia M McAndrew, Jason M Wilken, and Jonathan B Dingwell. Dynamic stability of human walking in visually and mechanically destabilizing environments. *Journal of biomechanics*, 44(4):644–649, 2011.

[96] Josh Merel, Matthew Botvinick, and Greg Wayne. Hierarchical motor control in mammals and machines. *Nature communications*, 10(1):1–12, 2019.

[97] Giorgio Metta, Lorenzo Natale, Francesco Nori, Giulio Sandini, David Vernon, Luciano Fadiga, Claes Von Hofsten, Kerstin Rosander, Manuel Lopes, José Santos-Victor, et al. The icub humanoid robot: An open-systems platform for research in cognitive development. *Neural networks*, 23(8-9):1125–1134, 2010.

[98] Theodore E Milner. Contribution of geometry and joint stiffness to mechanical stability of the human arm. *Experimental brain research*, 143(4):515–519, 2002.

[99] John G Milton, Toru Ohira, Juan Luis Cabrera, Ryan M Fraiser, Janelle B Gyorffy, Ferrin K Ruiz, Meredith A Strauss, Elizabeth C Balch, Pedro J Marin, and Jeffrey L Alexander. Balancing with vibration: a prelude for "drift and act" balance control. *PLoS One*, 4(10):e7427, 2009.

[100] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[101] Jun Morimoto and Kenji Doya. Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. *Robotics and Autonomous Systems*, 36(1):37–51, 2001.

[102] Ferdinando A Mussa-Ivaldi, Neville Hogan, and Emilio Bizzi. Neural, mechanical, and geometric factors subserving arm posture in humans. *Journal of Neuroscience*, 5(10):2732–2743, 1985.

[103] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE, 2018.

[104] Akira Nagamori, Christopher M Laine, and Francisco J Valero-Cuevas. Cardinal features of involuntary force variability can arise from the closed-loop control of viscoelastic afferented muscles. *PLoS computational biology*, 14(1):e1005884, 2018.

[105] Tarek Najjar and Osamu Hasegawa. Self-organizing incremental neural network (soinn) as a mechanism for motor babbling and sensory-motor learning in developmental robotics. In *International Work-Conference on Artificial Neural Networks*, pages 321–330. Springer, 2013.

[106] John R Napier. The prehensile movements of the human hand. *Bone & Joint Journal*, 38(4):902–913, 1956.

[107] Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive processing*, 12(4):319–340, 2011.

[108] Duy Nguyen-Tuong and Jan Peters. Online kernel-based learning for task-space tracking robot control. *IEEE transactions on neural networks and learning systems*, 23(9):1417–1425, 2012.

[109] Duy Nguyen-Tuong, Jan Peters, Matthias Seeger, and Bernhard Schölkopf. Learning inverse dynamics: a comparison. In *European symposium on artificial neural networks*, 2008.

[110] Chuanxin M Niu, Kian Jalaleddini, Won Joon Sohn, John Rocamora, Terence D Sanger, and Francisco J Valero-Cuevas. Neuromorphic meets neuromechanics, part i: the methodology and implementation. *Journal of neural engineering*, 14(2):025001, 2017.

[111] Lael U Odhner, Leif P Jentoft, Mark R Claffee, Nicholas Corson, Yaroslav Tenzer, Raymond R Ma, Martin Buehler, Robert Kohout, Robert D Howe, and Aaron M Dollar. A compliant, underactuated hand for robust manipulation. *The International Journal of Robotics Research*, 33(5):736–752, 2014.

[112] Katsuhiko Ogata. *Modern control engineering*. Prentice hall, 2010.

[113] Takayuki Osa, Jan Peters, and Gerhard Neumann. Hierarchical reinforcement learning of multiple grasping strategies with human instructions. *Advanced Robotics*, 32(18):955–968, 2018.

[114] Rieko Osu and Hiroaki Gomi. Multijoint muscle regulation mechanisms examined by measured human arm stiffness and emg signals. *Journal of neurophysiology*, 81(4):1458–1468, 1999.

[115] Thomas D O'Brien, Neil D Reeves, Vasilios Baltzopoulos, David A Jones, and Constantinos N Maganaris. Mechanical properties of the patellar tendon in adults and children. *Journal of biomechanics*, 43(6):1190–1195, 2010.

[116] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[117] Simone Parisi, Simon Ramstedt, and Jan Peters. Goal-driven dimensionality reduction for reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4634–4639. IEEE, 2017.

[118] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pages 2778–2787. PMLR, 2017.

[119] P Hunter Peckham and Jayme S Knutson. Functional electrical stimulation for neuromuscular applications. *Annu. Rev. Biomed. Eng.*, 7:327–360, 2005.

[120] Eric J Perreault, Robert F Kirsch, and Patrick E Crago. Effects of voluntary force generation on the elastic components of endpoint stiffness. *Experimental brain research*, 141(3):312–323, 2001.

[121] Eric J Perreault, Robert F Kirsch, and Patrick E Crago. Voluntary control of static endpoint stiffness during force regulation tasks. *Journal of neurophysiology*, 87(6):2808–2816, 2002.

[122] Josh I Petersen, Pranav Bhounsule, and Andy Ruina. Dynamic Walking MATLAB simulation guide. http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/ranger/ranger_paper/Reports/Ranger_Robot/control/simulator/doublependulum.html, 2008. [Online; accessed 19-July-2008].

[123] Veljko Potkonjak, Bratislav Svetozarevic, Kosta Jovanovic, and Owen Holland. The puller-follower control of compliant and noncompliant antagonistic tendon drives in robotic systems. *International Journal of Advanced Robotic Systems*, 8(5):69, 2011.

[124] Gill Andrews Pratt. Low impedance walking robots. *Integrative and Comparative Biology*, 42(1):174–181, 2002.

[125] Boris I Prilutsky. Muscle coordination: the discussion continues. *Motor Control*, 4(1):97–116, 2000.

[126] U Proske and DL Morgan. Muscle damage from eccentric exercise: mechanism, mechanical signs, adaptation and clinical applications. *The Journal of physiology*, 537(2):333–345, 2001.

[127] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

[128] Daniel Rasmussen, Aaron Voelker, and Chris Eliasmith. A neural model of hierarchical reinforcement learning. *PloS one*, 12(7):e0180234, 2017.

[129] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*. ICLR, 2017.

[130] John Rieffel, Francisco Valero-Cuevas, and Hod Lipson. Automated discovery and optimization of large irregular tensegrity structures. *Computers & Structures*, 87(5):368–379, 2009.

[131] Eric Rombokas, Evangelos Theodorou, Mark Malhotra, Emo Todorov, and Yoky Matsuoka. Tendon-driven control of biomechanical and robotic systems: A path integral reinforcement learning approach. In *2012 IEEE International Conference on Robotics and Automation*, pages 208–214. IEEE, 2012.

[132] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[133] Marco Santello, Matteo Bianchi, Marco Gabiccini, Emiliano Ricciardi, Gionata Salvietti, Domenico Prattichizzo, Marc Ernst, Alessandro Moscatelli, Henrik Jörntell, Astrid ML Kappers, et al. Hand synergies: integration of robotics and neuroscience for understanding the control of biological and artificial hands. *Physics of life reviews*, 17:1–23, 2016.

[134] Marco Santello, Martha Flanders, and John F Soechting. Postural hand synergies for tool use. *Journal of Neuroscience*, 18(23):10105–10115, 1998.

[135] Matthew Schlesinger and Domenico Parisi. Multimodal control of reaching-simulating the role of tactile feedback. *IEEE transactions on evolutionary computation*, 5(2):122–128, 2001.

[136] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[137] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[138] Sangok Seok, Albert Wang, Meng Yee Michael Chuah, Dong Jin Hyun, Jongwoo Lee, David M Otten, Jeffrey H Lang, and Sangbae Kim. Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot. *Ieee/asme transactions on mechatronics*, 20(3):1117–1129, 2014.

[139] Charles Scott Sherrington. Reflex inhibition as a factor in the co-ordination of movements and postures. *Experimental Physiology*, 6(3):251–310, 1913.

[140] Charles Scott Sherrington. Inhibition as a coordinative factor. *Nobelprize. org*, 1932.

[141] Lauri Stenroth, Jussi Peltonen, Neil J Cronin, Sarianna Sipilä, and Taija Finni. Age-related differences in achilles tendon properties and triceps surae muscle architecture in vivo. *Journal of Applied Physiology*, 113(10):1537–1544, 2012.

[142] Shinjiro Sueda, Andrew Kaufman, and Dinesh K Pai. Musculotendon simulation for hand animation. *ACM Transactions on Graphics (TOG)*, 27(3):83, 2008.

[143] Kuniyuki Takahashi, Tetsuya Ogata, Jun Nakanishi, Gordon Cheng, and Shigeki Sugano. Dynamic motion learning for multi-dof flexible-joint robots using active–passive motor babbling through deep learning. *Advanced Robotics*, 31(18):1002–1015, 2017.

[144] Jie Tan, Greg Turk, and C Karen Liu. Soft body locomotion. *ACM Transactions on Graphics (TOG)*, 31(4):26, 2012.

[145] Lena H Ting and J Lucas McKay. Neuromechanics of muscle synergies for posture and movement. *Current opinion in neurobiology*, 17(6):622–628, 2007.

[146] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE, 2012.

[147] Jennifer Brooke Treweek and Viviana Gradinaru. Extracting structural and functional features of widely distributed biological circuits with single cell resolution via tissue clearing and delivery vectors. *Current opinion in biotechnology*, 40:193–207, 2016.

[148] Francisco J Valero-Cuevas. *Fundamentals of neuromechanics*. Springer, 2016.

[149] Francisco J Valero-Cuevas, Vikrant V Anand, Anupam Saxena, and Hod Lipson. Beyond parameter estimation: extending biomechanical modeling by the explicit exploration of model topology. *IEEE Transactions on Biomedical Engineering*, 54(11):1951–1964, 2007.

[150] Francisco J Valero-Cuevas, BA Cohn, HF Yngvason, and Emily L Lawrence. Exploring the high-dimensional structure of muscle redundancy via subject-specific and generic musculoskeletal models. *Journal of biomechanics*, 48(11):2887–2896, 2015.

[151] Francisco J Valero-Cuevas and Marco Santello. On neuromechanical approaches for the study of biological and robotic grasp and manipulation. *Journal of NeuroEngineering and Rehabilitation*, 14(1):101, 2017.

[152] Alexander Verl, Alin Albu-Schäffer, Oliver Brock, and Annika Raatz. *Soft Robotics*. Springer, 2015.

[153] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

[154] Patrick M Wensing, Albert Wang, Sangok Seok, David Otten, Jeffrey Lang, and Sangbae Kim. Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots. *IEEE Transactions on Robotics*, 33(3):509–522, 2017.

[155] Yuxiang Yang, Ken Caluwaerts, Atil Iscen, Tingnan Zhang, Jie Tan, and Vikas Sindhwani. Data efficient reinforcement learning for legged robots. *arXiv preprint arXiv:1907.03613*, 2019.

[156] Tsuneo Yoshikawa. *Foundations of robotics: analysis and control*. MIT press, 1990.

[157] Felix E Zajac. Muscle and tendon properties models scaling and application to biomechanics and motor. *Critical reviews in biomedical engineering*, 17(4):359–411, 1989.

[158] Xiaodong Zhou and Shusheng Bi. A survey of bio-inspired compliant legged robot designs. *Bioinspiration & biomimetics*, 7(4):041001, 2012.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346 USA